

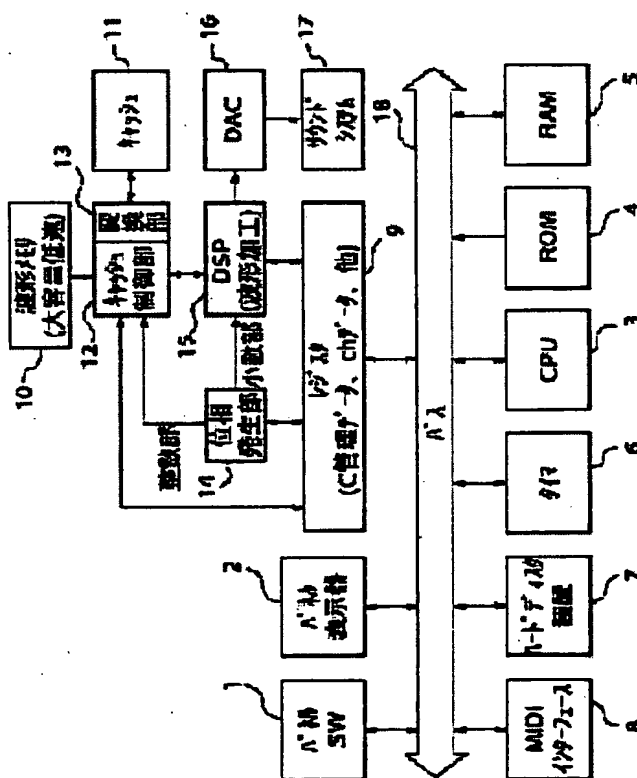
MUSICAL SOUND GENERATING DEVICE AND STORAGE MEDIUM

Patent number: JP2000276172
Publication date: 2000-10-06
Inventor: OKAZAKI MASATSUGU
Applicant: YAMAHA CORP
Classification:
 - international: G10H7/02; G06F12/08
 - european:
Application number: JP19990079523 19990324
Priority number(s): JP19990079523 19990324

Report a data error here

Abstract of JP2000276172

PROBLEM TO BE SOLVED: To generate a musical sound having the number of channels more than those limited by the slow data transfer speed of a storage device according to waveform data stored in the storage device.
SOLUTION: When a cache hit of waveform data whose sound generation is indicated is detected, a musical sound is generated by using waveform data held in a cache memory 11 and when a cache miss of the waveform data whose sound generation is indicated is detected, the waveform data regarding the sound generation indication are read out of a waveform memory 10 and transferred to the cache memory 11, so that a musical sound is generated by using the transferred waveform data. Further, when timbre switching is indicated, whether or not waveform data used preferentially for the switching-indicated timbre are held in the cache memory 11 is detected and when the waveform data are not held in the cache memory 11, the waveform data are read out of the waveform memory 10 and transferred to the cache memory 11.



Data supplied from the esp@cenet database - Worldwide



(19) The Japanese Patent Office

(12) Laid-Open Patent Application Publication (A)

(11) Laid-Open Patent Application Publication No. 2000-276172A

(43) Publication Date: October 6, 2000 (2002.10.6)

(51)

Int. Cl. ⁷	Classification Symbol	FI	Subject Code
G10H 7/02		G10H 7/00	521K 5B005
G06F 12/08		G06F 12/08	U 5D378

Request for Examination: not yet requested

Number of Claims: 7 OL (13 pages in total)

(21) Application No. Patent Application No. Hei 11-79523

(22) Application Date: March 24, 1999 (1999.3.24)

(71) Applicant: 000004075

Yamaha Corp.

No. 10-1, Nakazawa-cho, Hamamatsu-shi,
shizuoka-ken, Japan

(72) Inventor: Masashi OKAZAKI

c/o Yamaha Corp.,
No. 10-1, Nakazawa-cho, Hamamatsu-shi,
shizuoka-ken, Japan

(74) Agent: Patent Attorney, Toshihiko WATABE

F term (Reference): 5B005 JJ11 KH12 LL00 MM01
5D378 AD01 AD21 BB11 CC46

(54) [Title of the Invention] MUSIC SOUND GENERATOR DEVICE AND
RECORDING MEDIUM THEREOF

(57) [Abstract]

[Problem to be solved] To provide a recording medium and music sound generator device capable of generating music sounds on a number of channels greater than the number of channels limited by the data transfer speed of the storage device, based on waveform data stored

in storage device with a low data transfer speed.

[Solution] When a cache hit for waveform data specified for sound generation is detected, music sounds are generated using waveform data held in the cache memory 11; and when a cache miss for waveform data specified for sound generation instruction is detected, the waveform data specified for sound generation is read out from the waveform memory 10, transferred to the cache memory 11, and music sounds generated using the waveform data after transfer. When an instruction to switch tones is issued, a search is made to detect whether or not waveform data given priority by that switching instruction is held in the cache memory 11, and if not held in the cache memory 11, then that waveform data is read out from the waveform memory 10 and transferred to the cache memory 11.

[Claims]

[Claim 1] A music sound generator device comprising:

a low data transfer speed, first storage means for storing multiple waveform data;

a high data transfer speed, second storage means;

a decision means for deciding whether or not waveform data relating to a readout instruction is stored in the second storage means;

a transfer means for reading the waveform data relating to the readout instruction from the first storage means and transferring it to the second storage means according to the readout instruction, when the result from the decision means is that waveform data relating to the readout instruction is not stored in the second storage means; and

a music sound generation means for reading the stored waveform data from the second storage means, and generating music sounds based on the readout data,

wherein the transferring of waveform data by the transfer means and generation of music sounds by the music sound generation means are performed asynchronously.

[Claim2] A music sound generator according to claim 1, wherein the transfer means transfers the waveform data relating to the readout instruction to the second storage means in waveform data units.

[Claim 3] A music sound generator comprising:

a low data transfer speed, first storage means for storing multiple waveform data;

a high data transfer speed, second storage means;

a tone switching instruction means for commanding tone switching;

a decision means for determining whether or not the priority waveform data with a specified tone is stored in the second storage means, when tone switching is specified by the tone switching instruction means;

a transfer means for reading the waveform data from the first storage means and transferring it to the second storage means, when a result from the decision means is that priority waveform data is not stored in the second storage means; and

a music sound generation means for reading the stored waveform data

from the second storage means, and generating music sounds based on the readout waveform data.

[Claim 4] A music sound generator comprising:

- a low data transfer speed, first storage means for storing multiple waveform data including multiple waveform samples;

- a high data transfer speed, second storage means;

- a decision means for deciding whether or not waveform data relating to a sound generating instruction is stored in the second storage means;

- a transfer means for reading the waveform samples of waveform data relating to the sound generating instruction from the first storage means and transferring it to the second storage means, when a result from the decision means is that waveform data relating to the sound generating instruction is not stored in the second storage means; and

- a music sound generating means for sequentially reading the waveform samples for waveform data relating to the sound generating instruction from the second storage means according to the sound generating instruction, and generating music sounds based on the readout waveform samples;

wherein the transfer speed of waveform samples of waveform data from the transfer means is set higher than the readout speed of waveform samples from the music sound generating means.

[Claim 5] A recording medium storing a program capable of being executed on a computer and comprising:

- a decision module for deciding whether or not waveform data relating to the readout instruction is stored in the high data transfer speed, second storage means;

- a transfer module for reading the waveform data relating to the readout instruction, from the low data transfer speed, first storage means storing multiple waveform data, and transferring it to the second storage means according to the readout instruction, when results from the decision module show that there is no waveform data relating to the readout instruction in the second storage means; and

- a music sound generating module for reading waveform data from the second storage means, and generating music sounds based on the readout

waveform data,

wherein the transferring of waveform data by the transfer module and the generating of music sounds by the music sound generation module are performed asynchronously.

[Claim 6] A recording medium storing a program capable of being executed on a computer and comprising:

a tone switching instruction module for instructing tone switching;

a decision module for deciding whether or not the priority waveform data relating to an instructed tone is stored in the high data transfer speed, second storage means, when there is a tone switching instruction from the tone switching instruction module;

a transfer module for reading the waveform data from the low data transfer speed, first storage means storing multiple waveform data, and transferring it to the second storage means, when the result from the decision module is that the priority waveform data is not stored in the second storage means; and

a music sound generation module for reading the stored waveform data from the second storage means, and generating music sounds based on the readout waveform data.

[Claim 7] A recording medium storing a program capable of being executed on a computer and comprising:

a decision module for deciding whether or not waveform data relating to the sound generating instruction is stored in the high data transfer speed, second storage means;

a transfer module for reading waveform samples of waveform data relating to the sound generating instruction from a low data transfer speed, first storage means storing multiple units of waveform data including multiple waveform samples according to the sound generating instruction, and transferring it to a second storage means when the result from the decision module is that there is no waveform data relating to the sound generating instruction in the second storage means; and

a music sound generating module for sequentially reading the waveform samples for waveform data relating to the sound generating instruction from the second storage means according to the sound generating instruction, and generating music sounds based on the

readout waveform samples;

wherein the transfer speed of waveform samples of waveform data from the transfer module is set higher than the readout speed of waveform samples from the music sound generating module.

[Detailed Description of the Invention]

[0001]

[Field of the Invention]

The present invention relates to a recording medium and music sound generator device capable of generating music sounds on a number of channels greater than the number of channels limited by the data transfer speed of the storage device, based on waveform data stored in storage device with a low data transfer speed.

[0002]

[Background Art]

The following types of music sound generator devices are utilized in the background art for generating music sounds based on waveform data stored in a storage device (including waveform memories (ROM)) with a low data transfer speed. Namely,

(1) Music sound generator devices for generating a target music sound waveform by interpolation of a waveform sample read from the waveform memory, storing the read waveform sample utilized for interpolation in the cache memory, and generating a waveform based on the waveform sample in the cache; and

(2) Music sound generator devices for loading waveform data stored in a hard disk which is a storage device with a low transfer speed, into a low-capacity, high-speed RAM, and generated music sounds based on this loaded waveform data.

[0003]

[Problem to be Solved by the Invention]

However, among the devices of the background art, in the music sound generator device (1), as generating of the music sound waveform progresses, when the waveform sample for generating the music sound waveform by interpolation is not being held in the cache memory, that waveform sample cannot be read out from the waveform memory so the number of music sound waveform channels that are generated depends on the waveform memory data transfer speed.

[0004]

Among the devices of the background art, in the music sound generator device (2) when an instruction to generate a music sound is issued, that music sound could not be generated if the waveform data for that

music sound was not loaded in the RAM.

[0005]

In view of the above problems, the present invention has the object of providing a recording medium and music sound generator device capable of generating music sounds on a number of channels greater than the number of channels limited by the data transfer speed of the storage device, based on waveform data stored in storage device with a low data transfer speed.

[0006]

[Means for Solving the Problem]

In order to achieve the above objectives, the music sound generator device of claim 1 includes: a low data transfer speed, first storage means for storing multiple waveform data; a high data transfer speed, second storage means; a decision means for deciding whether or not waveform data relating to a readout instruction is stored in the second storage means; a transfer means for reading the waveform data relating to the readout instruction from the first storage means and transferring it to the second storage means according to the readout instruction, when the result from the decision means is that waveform data relating to the readout instruction is not stored in the second storage means; and a music sound generation means for reading the stored waveform data from the second storage means, and generating music sounds based on the readout data, wherein the transferring of waveform data by the transfer means and generation of music sounds by the music sound generation means are performed asynchronously.

[0007]

The first storage means here is typically a waveform memory (ROM); however, the first storage means is not limited to a waveform memory and may be any storage means with a low data transfer speed such as a hard disk or an MD. The second storage means is typically a cache memory but is not limited to a cache memory, and if the data transfer speed is fast, it may even be a RAM. The above holds true even for different claims.

[0008]

The transfer means preferably transfers waveform data relating to the readout instruction to the second storage means in waveform data

units.

[0009]

The music sound generator device according to claim 3 includes a low data transfer speed, first storage means for storing multiple waveform data; a high data transfer speed, second storage means; a tone switching instruction means for commanding tone switching; a decision means for determining whether or not the priority waveform data with a specified tone is stored in the second storage means, when tone switching is specified by the tone switching instruction means; a transfer means for reading the waveform data from the first storage means and transferring it to the second storage means, when a result from the decision means is that priority waveform data is not stored in the second storage means; and a music sound generation means for reading the stored waveform data from the second storage means, and generating music sounds based on the readout waveform data.

[0010]

The music sound generator device according to claim 4 includes a low data transfer speed, first storage means for storing multiple waveform data including multiple waveform samples; a high data transfer speed, second storage means; a decision means for deciding whether or not waveform data relating to a sound generating instruction is stored in the second storage means; a transfer means for reading the waveform samples of waveform data relating to the sound generating instruction from the first storage means and transferring it to the second storage means, when a result from the decision means is that waveform data relating to the sound generating instruction is not stored in the second storage means; and a music sound generating means for sequentially reading the waveform samples for waveform data relating to the sound generating instruction from the second storage means according to the sound generating instruction, and generating music sounds based on the readout waveform samples; wherein the transfer speed of waveform samples of waveform data from the transfer means is set higher than the readout speed of waveform samples from the music sound generating means.

[0011]

In order to achieve the above objects, the recording medium of claim

5, includes a decision module for deciding whether or not waveform data relating the readout instruction is stored in the high data transfer speed, second storage means; a transfer module for reading the waveform data relating to the readout instruction, from the low data transfer speed, first storage means storing multiple waveform data, and transferring it to the second storage means according to the readout instruction, when results from the decision module show that there is no waveform data relating to the readout instruction in the second storage means; and a music sound generating module for reading waveform data from the second storage means, and generating music sounds based on the readout waveform data, wherein the transferring of waveform data by the transfer module and the generating of music sounds by the music sound generation module are performed asynchronously.

[0012]

The recording medium of claim 6, includes: a tone switching instruction module for instructing tone switching; a decision module for deciding whether or not the priority waveform data relating to an instructed tone is stored in the high data transfer speed; second storage means, when there is a tone switching instruction from the tone switching instruction module; a transfer module for reading the waveform data from the low data transfer speed, first storage means storing multiple waveform data, and transferring it to the second storage means, when the result from the decision module is that the priority waveform data is not stored in the second storage means; and a music sound generation module for reading the stored waveform data from the second storage means, and generating music sounds based on the readout waveform data.

[0013]

The recording medium of claim 7, includes a decision module for deciding whether or not waveform data relating to the sound generating instruction is stored in the high data transfer speed, second storage means; a transfer module for reading waveform samples of waveform data relating to the sound generating instruction from a low data transfer speed, first storage means storing multiple units of waveform data including multiple waveform samples according to the

sound generating instruction, and transferring it to a second storage means when the result from the decision module is that there is no waveform data relating to the sound generating instruction in the second storage means; and a music sound generating module for sequentially reading the waveform samples for waveform data relating to the sound generating instruction from the second storage means according to the sound generating instruction, and generating music sounds based on the readout waveform samples; wherein the transfer speed of waveform samples of waveform data from the transfer module is set higher than the readout speed of waveform samples from the music sound generating module.

[0014]

[Embodiment of the Invention]

The modes of the present invention are described hereinafter in detail with reference to the accompanying drawings.

[0015]

FIG. 1 is a block diagram showing the overall structure of the music sound generator device of the embodiment of the invention.

[0016]

The music sound generator device of the embodiment of this invention as shown in this same figure includes: a panel switch 1 containing multiple switches for entering different types of information; a display device 2 for displaying various different types of information and having for example a large liquid crystal display (LCD) and a light emitting diode (LED); a CPU 3 for controlling the entire device, a ROM 4 for storing different types of table data and control programs executed by the CPU 3; a RAM 5 for temporarily storing performance data, all types of information entries, and processing results; a timer 6 for counting various time and the interrupt time for the timer interrupt processing; a hard disk device 7 for storing different types of data and different types of application programs including the control programs; an MIDI (musical instrument digital interface) 8 for entering MIDI signal from external sections and outputting MIDI signals to external sections; and a register 9 containing areas for storing data such as ch (channel) data (described later on using FIG. 2) and C (cache) processing data (described later

on using FIG. 3). These components 1 through 9 are mutually connected via a bus 18.

[0017]

As described above, the hard disk 7 is capable of storing a control program for the CPU 3 to execute. When there is no control program stored in the ROM 4, the control program is stored onto this hard disk, and loaded by the RAM 5 to make the CPU 3 perform the same operation as when the control program is stored in the ROM 4. The above construction allows easily adding control programs and making version upgrades.

[0018]

The MIDI interface 8 is not limited to a dedicated interface, and may for example be a general-purpose interfaces such as an RS-232C or USB (universal serial bus), IEEE1394 (I triple-e 1394). In that case, data other than the MIDI message can be simultaneously sent and received.

[0019]

The register 9 is connected to: a cache controller 12 for loading the waveform data stored in the large capacity (for example 128 Mword) and low data transfer speed (for example 6 Mword per second) waveform memory (ROM) 10 into a small capacity (for example 4 Mword) and high data transfer speed (for example 66 Mword per second) cache memory 11 and for reading the waveform data loaded in the cache memory 11; a phase generator 14 for generating a readout address (relative address) of waveform data for each music sound generator channel by accumulating the F numbers FN showing the quantitative progress per one sample during readout of the waveform data for each channel; and a DSP (digital signal processor) 15 for processing waveform data read out from the cache memory 11.

[0020]

When generating music sounds, the music sound generator device of the embodiment of this invention here constantly reads waveform data from the cache memory 11 that serves as a reference for generating music sounds, and generates music sounds based on this waveform data. When this waveform data serving as the reference for generating music sounds is not being held in the cache memory 11, or in other words,

when a cache miss has occurred, that waveform data is read from the waveform memory 10, and after being transferred to the cache memory 11, is read from the cache memory 11 to generate music sounds. At this point, the transfer speed of the waveform data from the waveform memory 10 to the cache memory 11 is set to be faster than the quantitative progress (readout speed) for the readout address of the waveform data on the cache memory 11 so that there is no incomplete transfer of the waveform samples to be loaded to the cache memory 11.

[0021]

In this embodiment, the device sampling frequency is for example set to 50 kHz. When the readout address generated by the phase generator 14 contains a decimal fraction, the waveform sample of that address is calculated by direct interpolation so that two waveform samples must be read out per single period and therefore the actual sampling frequency is 100 kHz.

[0022]

Music sounds for one channel can be generated if one waveform sample (The data capacity is assumed to be 1 word) is read out in one period of this 100 kHz sampling frequency. Therefore, when the stored waveform data (waveform sample) is read and music sounds generated in this embodiment, by continually accessing the cache memory 11, music sounds for $66 \text{ Mword/sec./channel} \div 100 \text{ kword/sec./channel} = 660$ channels) can be generated.

[0023]

In this embodiment however, data transfer is performed at a maximum of 6 Mword per second within one sampling period. In other words, since waveform data is also transferred from the waveform memory 10 to the cache memory 11, then 60 channels must be subtracted from the above 660 channels since $6 \text{ Mword/sec.} \div 100 \text{ kword/sec./channel} = 60$ channels. The music sound generator device of this embodiment is therefore capable of generating a maximum of 600 channels.

[0024]

The phase generator 14 generates the readout address of a waveform data (real number) for each channel based on channel data for each sound generating channel (The number of channels is a maximum of 600

channel as described above) stored in the register 9. The phase generator supplies this integer portion of this number to the cache controller 12 and the decimal fraction portion to the DSP 15.

[0025]

The cache controller 12 decides whether or not the waveform data to be read for each channel is stored in the cache memory 11, based on the C control data stored in the register 9. When the waveform data is not stored, the cache controller 12 starts reading waveform data on the applicable channel and simultaneously reads that waveform data from the waveform memory 10 and transfers it to the cache memory 11. However, when the waveform data is stored, the cache controller 12 reads out waveform data for the position (and that next position) corresponding to the integer portion of the read out address generated on the applicable channel, and supplies it to the DSP 15.

[0026]

The waveform data here is read from the waveform memory 10 in block units (1 kWord is one block in this embodiment) and loaded into the cache memory 11. The waveform data capacity to be read out is usually larger than 1 kWord so that multiple blocks are loaded into the cache memory 11. However, when the cache memory 11 does not possess the total available area capacity for these multiple blocks, then the waveform data for each block is stored discretely in the cache memory 11. The phase generator 14 on the other hand, consecutively issues the loaded addresses as addresses for the applicable waveform data. A virtual address is therefore defined for the actual address on the cache memory 11, and by repositioning this virtual address (virtual address for each block), the waveform data addresses that are dispersed actual addresses, are seen as consecutive objects. This type of technology is called garbage collection. This technology eliminates the need to transfer data stored in the cache memory 11 when collectively using the empty available area and therefore allows high speed processing.

[0027]

In order to read out the target waveform data from the cache memory 11, the cache controller 12 connects to the cache memory 11 via the address converter 13 for converting this virtual address into an

actual address on the cache memory 11.

[0028]

The DSP 15 performs sound generation channel processing, effect processing, and mixer processing. The DSP 15 outputs the digital music sound data from this processing to the DAC (digital to analog) converter 16. The DAC 16 converts the digital audio data supplied from the DSP 15, into an analog sound generation signal, and converts this signal into sound via a sound system 17 comprised of an amplifier and speaker, etc.

[0029]

The sound generation channel processing here mainly includes for example, sample interpolation processing, tone filter processing, volume envelope processing, and channel accumulation processing.

[0030]

In sample interpolation processing, interpolation processing is performed based on the decimal fraction portion of the readout address supplied from the phase generator 14 and the waveform data supplied from the cache controller 12, and generates a waveform sample for the interpolated waveform data. In the present embodiment, the cache controller 12 provides two waveform samples per channel to the DSP 15, and the DSP 15 performs interpolation processing to generate interpolation samples based on these two waveform samples. Of course, interpolation can be performed to a higher degree based on the newly readout waveform samples and previously readout waveform samples.

[0031]

In the tone filter processing, the tone is controlled by subjecting the interpolated waveform data to processing such as with a low-pass filter. This low pass filter processing is performed based on the supplied parameters such as the cutoff characteristics and the resonance characteristics, etc.

[0032]

In volume envelope processing, a time change (envelope) in the volume from the rise to the fall of the music sound is applied to the waveform data after tone filter processing. This volume envelope processing is also performed based on EG (envelope generator) control information supplied as a parameter.

[0033]

In the channel accumulating process, the waveform data applied with an envelope, is regulated at the respective for the effect processing levels (in the embodiment, 3 types of processing, including the chorus processing, reverb processing, and equalizer processing) and accumulative channel processing performed on all channels before output to the respective effect processing.

[0034]

In the chorus processing, a chorus effect is applied to the waveform data output for the applicable process.

[0035]

In the reverb processing, a reverberation effect is applied to the waveform data output for the applicable process.

[0036]

In the mixer processing, the respective volume levels for the waveform data applied with the chorus effect and waveform data applied with the reverb effect are regulated, and the volume controlled waveform data and waveform data output for equalizer processing are subjected to mixing.

[0037]

In the equalizer processing, the frequency characteristics of the waveform data mixed in the mixer processing, are processed.

[0038]

The DSP 15 supplies this waveform data whose frequency characteristics were processed, to the DAC 16.

[0039]

The phase generator 14 and the DSP 15 must therefore time divide the processing for generating music sounds for a maximum of 600 channels within one sampling period. To achieve this processing, each of the circuits may be placed in parallel or a pipeline configuration utilized as needed. In other words, a pipeline configuration is employed to handle the processing for the time divided 600 channels in multiple steps. A pipeline configuration of two parallel pipelines can each handle 300 time divided channels in multiple steps, or a configuration of four pipelines in parallel can handle 150 time divided channels in multiple steps. The number

of pipeline stages can generally be reduced by increasing the number of circuits in parallel.

[0040]

FIG. 2 is a diagram showing the data format of the channel data stored in the register 9.

[0041]

The channel data is made up multiple units of data needed for generating music sounds on the applicable sound generating channels. The register 9 contains areas for channel 1 to channel 600 where data for 600 channels is stored. The channel data formats are all jointly used so the data format for the channel data on channel 3 is given as a typical example.

[0042]

The channel (ch) data in the same figure is made up of: an F number FN; the overall volume for the generated music sounds; a waveform ID for specifying the waveform data in the waveform memory 10 for utilizing the generated music sounds; relative address information for a specified position (for example, the readout start position, loop readout start position and loop readout end position) as the basis for the lead storage position (address) of the waveform data of the applicable waveform (matching the waveform ID) on the waveform memory 10 (or the cache memory 11); the DCF (digital controlled filter) control information (including cutoff characteristics and resonance characteristics) as parameters for filter processing in the tone filter processing; EG control information as a parameter used in volume control in the volume envelope processing; other information utilizing in generating music sounds; and on/off data showing the note-on/off states respectively as "0" and "1".

[0043]

The F number FN shows the address readout speed per one sample when reading the waveform data. This value is determined by establishing the pitch of the music sound to generate and the waveform data used to generate that sound. The tone emitter 14 is synchronized with the system sampling frequency (50 KHz in this embodiment as described above), the F number FN is accumulated at each channel and, a readout address (relative address) generated for waveform data of each

channel.

[0044]

ID is respectively assigned to each of the waveform data stored in the waveform memory 10. The target waveform data can then be specified on the waveform memory 10 by specifying the waveform ID within that channel.

[0045]

Generating of music sounds from on/off data starts for the applicable channel, at the point in time that the value changes from a "0" to a "1" and an instruction is sent to the DSP 15 to commence release of music sounds on the applicable channel.

[0046]

FIG. 3 is a diagram showing the format of the C control data stored in the register 9.

[0047]

The C control data includes multiple units of data for managing the waveform data stored in the cache memory 11. Areas cw1 through cw600 are formed in the register 9 for storing data for 600 waveforms. The channel data formats for each of the C control data are all jointly used and so the data format for the control data of cw (cache wave) 3 is given as a typical example.

[0048]

The C control data in the same figure is made up of: a waveform ID for applying the same effect as the waveform ID within the channel data; a transfer state TS showing the transfer status of the applicable waveform data to the cache memory 11; the transfer speed TA showing the speed when transferring the applicable waveform data to the cache memory 11; a cache storage position virtual address for expressing the virtual address of the storage position on the cache memory 11 on the applicable waveform data; a transfer end size for showing the data size of the data transferred to the cache memory 11 from among the applicable waveform data; a total size for showing the overall size of the applicable waveform data; a waveform memory storage position linear address for showing storage position of the applicable waveform data on the waveform memory 10 with a real address (a linear address since the waveform data is usually (consecutively)

stored linearly on the waveform memory 10).

[0049]

The transfer state TS acquires an integer value for any from "0" to "2" and these integer values signify the following.

[0050]

0: signifies a state where waveform data showing the applicable waveform ID is not used in generating music sounds.

1: signifies a state where waveform data showing the applicable waveform ID is currently being transferred in the cache memory 11.

2: signifies a state where transfer of waveform data showing the applicable waveform ID to the cache memory 11 has been completed.

The transfer speed TA utilizes any integer value from "0" to "120" and the number of blocks transferred per second (in this embodiment, 50 blocks equals 50 kwords) is set as one unit. A TA of 0 for example shows a state where no transfer of data to the cache memory 11 is performed. A TA of 1 shows a state where data transfer to the cache memory 11 is performed at 50 blocks per second. A TA of 2 shows a state where data transfer to the cache memory 11 is performed at 100 blocks per second. Therefore, the larger the value set as the transfer speed TA, the faster the transfer speed of a number of blocks per second.

[0051]

Here, the number of blocks transferred per second is set as 50 blocks because this transfer speed is equivalent to the speed for reproducing music sounds (when reading waveform samples for waveform data at the system sampling frequency (= 50 kHz)) from waveform data when the F number FN was set to "1".

[0052]

The transfer speed TA can utilize up to a maximum of "120" as described next. This is because the data transfer speed from the waveform memory 10 is 6 Mwords per second in this embodiment. Therefore, when there are multiple units of data for transfer to the cache memory 11, then "120" is used to distribute this data. (In other words, the total value of the transfer speed TA assigned to each load does not exceed "120".) The following methods may be used for distributing this data. Namely,

(1) A method where an integer value larger than the load during tone switching is assigned with priority to the load during sound generation;

(2) A method where a number of a value equal to or higher than the integer is assigned to the load during sound generation according to the F number FN of the channel for generating sounds using that waveform data; and

(3) A method where an integer value less than 120 may be assigned to the load during tone switching (For example, when the transfer speed TA assigned to other loads has reached "120" then a transfer speed TA of "0" may be assigned to the load during the remaining tone switching.)

In the method in (1), the load for generating sounds is the applicable waveform data to be read out from the waveform memory 10 and loaded into the cache memory 11 when waveform data for sounds yet to be generated have still not been stored in the cache memory 11. The load during tone switching in this embodiment, is the applicable waveform data read out from the waveform memory 10 and loaded into the cache memory 11 for example in case that waveform data given priority for the specified tone (for example, the most frequently used) is not stored in the cache memory 11 when the data instructing the tone switching is supplied as performance data.

[0053]

In the above method (2), the integer value according to the F number FN is for example an F number FN with numbers below the decimal point rounded off so that for example a "2" is generated when the FN equals 1.5. Setting an integer equal or larger than the integer according to the F number FN serves to set one unit of transfer speed as the reference for the transfer speed required when the F number FN equals 1. When the F number FN is set as a larger value than "1", unless waveform data is transferred to the cache memory 11 at a transfer speed based on this reference unit, then the readout speed of the address might exceed that of addresses for waveform samples that were already transferred. If this happens, then music sounds cannot be generated.

[0054]

Unlike the method (2), the method (3) contains no conditions for setting the values of transfer speed TA during the load in tone switching such as used in method (2). These conditions are not set because there is a time surplus from the time of the tone switching instruction to the time the music sounds are actually generated with that tone.

[0055]

The cache storage position virtual address as described above, expresses the virtual address of the storage position on the cache memory 11 holding the applicable waveform data. The blocks including the applicable waveform data discretely placed on the real address space on the cache memory 11, are again placed in consecutive positions on the virtual address space. In the present embodiment, each waveform sample of the applicable waveform data is read based on the cache storage position virtual address so the phase generator 14 can read and issue the address just by accumulating the F numbers FN. The cache controller 12 automatically repositions the data block on the virtual address space, and based on these results, the cache controller 12 rewrites the address conversion table in the address converter 13 for mutually converting the virtual address and the real address. This control processing executed by the cache controller 12 is already shown in detail in JP-A Hei 6-35473 and so the description is omitted here. By utilizing the hardware type address conversion disclosed in this document, access to the cache memory 11 by way of the virtual address can be achieved without a time delay.

[0056]

After first describing the general concept of the control processing executed by the music sound generator device structured as described above, a detailed description is related with reference to FIG. 4 through FIG. 7.

[0057]

The music sound generator device of this embodiment performs the following main processes, namely the music sound generator device: (1) detects a cache hit or a cache miss for waveform data specified for generating a sound, and when a cache hit is detected, a music sound is generated using the waveform data held in the cache memory

11; and when a cache miss is detected, the waveform data relating to the specified sound to generate is read from the memory 10, and transferred to the cache memory 11 and the music sound is generated using the waveform data after transfer;

(2) when tone switching is instructed, detects whether or not priority waveform data at a tone specified for switching is being held in the cache memory 11. When that waveform data is not being held in the cache memory 11, that waveform data is read from the waveform memory 10 and transferred to the cache memory 11;

(3) transfers waveform data from the waveform memory 10 per (1) and (2) above, to the cache memory 11, and asynchronously reads the waveform data (waveform sample) held in the cache memory 11 for generating music sounds for each channel;

(4) performs caching of waveform data into the cache memory 11 in single waveform units (however, the transfer of waveform data in single waveform units from the cache memory 11 is not required and in this embodiment, the waveforms are transferred by time division in multiple portions arrayed in parallel. In other words, when cache management such as load instructions for when there were only hits or when a cache hit was identified, is executed only in units of a single waveform and not for each sample or in multiple sample units within a single waveform. The waveform data may also be stored discretely on the cache memory 11 or may be stored consecutively.)

(5) transfers waveform data during a cache miss per (1) above or in other words, transfers data from the waveform memory 10 to the cache memory 11 at a faster speed than the waveform sample readout speed during generation of music sounds.

The control processing for (1) through (5) is next described in detail.

[0058]

FIG. 4 is a flow chart showing the sequence for the tone switching event processing executed by the music sound generator device of this embodiment and in particular by the CPU3 for implementing the control processing (2). This processing starts up when tone switching is supplied at the performance data.

[0059]

In the same figure, the tone number (new tone number) for the new tone instructed by an instruction from the tone change event is acquired and stored in the area TC (hereafter, the contents of this area are referred to as "tone TC") maintained at a specified position on the RAM 5 (step S1).

[0060]

The waveform IO for the waveform data loaded in the cache memory 11 according to the priority level within the different types of waveform data used in the tone TC is next established (step S2).

[0061]

A decision is next made whether waveform data matching this established waveform ID should be newly loaded in the cache memory 11 (step S3). Whether or not waveform data should be newly loaded is decided as described next.

[0062]

In other words, a check is made whether a waveform ID identical to the waveform ID established in step S2 is present in C control data whose transfer state TS is other than "0". If not present, then the waveform data matching that waveform ID must be newly loaded.

[0063]

In step S3, when there is no waveform data to be newly loaded because the target waveform data is being transferred to the cache memory 11 or transfer is completed, the tone change event processing is promptly terminated. However, when there is waveform data to be newly loaded, the processing subroutine (described later using FIG. 6) for executing instructions to load the waveform data into the cache memory 11 is executed (step S4) and then this tone change event processing is terminated.

[0064]

This load instruction processing subroutine is structured to allow switching to low speed loading or high speed loading in one subroutine according to the value set in the transfer speed TA as described later on. In step S4, the low speed instruction is performed by setting a value corresponding to "low speed" in the transfer speed TA. A low speed is set because there is no need for high speed loading during the tone switching.

[0065]

FIG. 5 is a flow chart showing the sequence for note-on event processing that is executed by the CPU 3. This sequence implements the control process (1). This processing is started when performance data is supplied as the note-on event.

[0066]

In the same figure, the note number and (&) the velocity are acquired from the supplied note-on event, and these are respectively stored in the area NN (hereafter the contents of this area are called ("note number NN") maintained in a specified position on the RAM 5, and an area VEL (hereafter the contents of this are called "velocity VEL") maintained in a specified position on the RAM 5 (step S11).

[0066]

Next, the waveform ID matching the note number NN and the tone TC (this tone TC is already set when the note-on event is supplied) is detected (step S12).

[0068]

A decision is next made whether the waveform data for this detected waveform ID is loaded in the cache memory 11 or not (step S13). This decision is made by the same method as used in step S3.

[0069]

In step S13, when the target waveform data is not loaded in the cache memory 11, a load instruction process subroutine (however, a high-speed instruction) the same as in step S4 is executed (step S14). Then, the process proceeds to step S15, when the target waveform data is loaded in the cache memory 11, step S14 is skipped and operation proceeds to step S15.

[0070]

In step S15, an available channel is detected and sounds are assigned to those channels.

[0071]

Proceeding to step S16, the different parameters are set according to the tone TC and the note number NN on the assigned channels. More specifically, the F number FN, the waveform ID, the relative address, the DCF control information, the EG control information and other information are written into the channel data area corresponding to

the assigned channel among the channel data areas for the 600 channel in the register 9.

[0072]

Then, after the note-on was set in the applicable channel (step S17), this note-on even processing is terminated. The note-on settings here are made by setting the note-on/off data in the channel data area of S16, to a note-on ("1").

[0073]

FIG. 6 is a flow chart showing in detail the procedure for the load instruction processing subroutine of step S4 and S14. This load instruction processing executes the control process (5).

[0074]

First of all in this figure, the minimum k (k is any integer from 1 to 600) is acquired from the areas cwk where the transfer state TS is "0" among the C control data storage areas for 600 waveforms of register 9 (step S21).

[0075]

Next areas on the cache memory 11 are assigned for storing waveform data showing the established (or detected) waveform ID (step S22). These areas are for example assigned as follows. Namely,

- (1) Multiple waveform data stored in the cache memory 11 are filled into the virtual address;
- (2) Then, an area for writing the waveform data indicated by the waveform ID is secured in the available area; and
- (3) If the available area is inadequate then areas storing currently unused waveform data are released for use, and the processes in (1) and (2) are repeated.

[0076]

The required parameters are next set in the area cwk in step S23. More specifically, the waveform ID, transfer state TS, cache storage position virtual address, total size and waveform memory storage position linear address are written.

[0077]

This loading instruction process then simultaneously terminates after the transfer speed TA for each waveform data is assigned (distributed) in step S24 according to the priority level of the

waveform data for loading. This distribution is performed by the method described above and each of the distributed transfer speeds TA are written into the corresponding transfer speed TA area within the applicable area cwk. A transfer speed equivalent to a "low speed" is in this way set in the tone change event processing, and a transfer speed equivalent to a "high speed" is in this way set in the note-on event processing. The "low speed" and "high speed" are relative expressions for transfer speeds in the tone change event processing and the note-on event processing; and signify that a larger speed is assigned for transfer in the note-on event processing as a result of the distribution. Even in cases where set to "low speed", a "120" is assigned as the transfer TA speed if other transfers are not being performed.

[0078]

FIG. 7 is a flow chart showing the transfer processing procedure executed by the cache controller 12 that also implements the control processes (3) and (4). This transfer process starts when the load instruction is issued by the load instruction processing subroutine in FIG. 6.

[0079]

In FIG. 7, the minimum k (default value) in the area cwk specified by the load (TS=1) is first of all detected, and stored in the region L (hereafter, these contents are called "index L") maintained in a specified position on the RAM 5 (step S31).

[0080]

Then the transfer speed TA (L) within the region cwL indicated by the index L is acquired, and these contents are stored in the software counter area.CNT (hereafter these contents are called "counter CNT") maintained in a specified position on the RAM 5 (step S32).

[0081]

The waveform data corresponding to the waveform ID (L) within the area cwL indicated by the index L is next read out from the waveform memory 10, and transferred to an area assigned previously in step S22 on the cache memory 11 (step S33). This transfer is performed in single or multiple block portions or in other words, in blocks where a transfer speed of 50 blocks per second can be obtained.

[0082]

In the next step S34, a decision is made whether or not transfer of waveform data corresponding to the waveform ID (L) was completed. This decision is made by comparing the size of completed transfers within the area cwL versus the total size.

[0083]

When the transfer of waveform data matching the waveform ID (L) was not completed in step S34, the counter CNT is decremented just by "1" (step S35) and then a decision made on whether the counter CNT is "0" (step S36).

[0084]

In step S36, when the CNT \neq 0, there is a block of waveform data to be transferred, so returning to step S33, the transfer of one or multiple blocks of data is repeated.

[0085]

However in step S34, when the transfer of waveform data matching waveform ID (L) has been completed, the transfer speed (L) within the area cwL is set to "2" (step S37), and a decision then made on whether there are other areas cwk with a transfer state TS of "1" or not (step S38).

[0086]

In step S38, when there are no other areas cwk where the transfer state TS is "1", this transfer process terminates. However, when there are other areas cwk where the transfer state TS is "1", then the transfer speed TA is re-allotted among these areas cwk (step S39). This re-allotment or redistribution for example, takes into account the priority level and causes of triggering among the waveform data being transferred. The transfer speed TA re-allotted in this way, is written into the matching transfer speed TA area.

[0087]

Next, in step S40 to find the waveform to transfer next, the next k (the next "k" is here selected by a circulating method. For example, when the area where the transfer state TS is "1", is k = 2, 5, 10, 25, if the "L" of cwL is "2", then the next "k" = 5, if the "L" is "25", then the next "k" is "2".) of the area cwL that was transferred in the previous stream of areas cwk where the transfer state TS is

"1", is acquired and is set in the next index L, and the transfer speed TA (L) within the area cwl shown by the index L is acquired the same as in step S32, and after storing it in the counter CNT (step S41) the process returns to step S33.

[0089]

In step S36 on the other hand, when CNT = 0, there is no block of waveform data to be sent so the process proceeds to step S40.

[0089]

When generating music sounds in this embodiment, the waveform data is in this way supplied at a high speed by utilizing just the waveform data stored in the cache memory so that the problem of a limit on the number of music sound generating channels due to delays in the transfer speed of the waveform memory is eliminated.

[0090]

When there is a cache hit, the music sounds are generated using waveform data held in the cache memory so there is no need to transfer waveform data from the waveform memory to the cache memory.

[0091]

When there is a cache miss, for waveform data specified for generating music sounds, the waveform data is transferred to the cache memory at a higher priority than other waveform data specified for loading due to other causes. Moreover the lower limit of this transfer speed is controlled according to the F number of the applicable waveform data and therefore eliminates the state where waveform data for loading out from the cache memory during generation of music sounds is not transferred, and so the present invention can achieve stable generation of music sounds.

[0092]

The readout progress of waveform sample addresses is not synchronized with the transfer of waveform data from the waveform memory into the cache memory so that there are no restrictions on the method used for transferring waveform data to the cache memory. Therefore, efficient transfer methods such as block transfer and burst transfer can be utilized.

[0093]

The caching of waveform data into the cache memory is carried out

in single waveform units so that data management is simple since for example, methods are not utilized that distribute the waveform data such that only a portion of the data is loaded into the cache.

[0094]

Data possessing a high priority due to tone selection is stored beforehand into the cache memory so that cache hits of the first waveform data for use can be obtained with good efficiency.

[0095]

In the present embodiment, a waveform memory (ROM) was utilized as the storage device for storing waveform data. However, the storage device is not limited to this and, for example MD devices and hard disk devices capable of random access may be utilized. In that case, first of all, just the lead portions of all the waveform data are stored into the cache memory, and then the waveform to be used and necessary waveforms are transferred in sequence to the cache memory from the applicable storage device.

[0096]

The present embodiment is structured to load high priority waveform data during tone selection. However, this invention is not limited to that method and for example the waveform data to be used next may be predicted according to changes such as tempo and the performance sound range, and this predicted waveform data then specified in advance for its time of use and that specified waveform data may then be loaded in advance.

[0097]

Also in the present embodiment, a so-called full associative method was employed when reading waveform data from the waveform memory into an optional available location in the cache memory. However, this invention is not limited to that method and, either the direct mapping method or the set associative methods may be utilized.

[0098]

Needless to say, the object of the present invention is achieved by supplying the software program code for implementing the embodiment and stored on a recording medium to the system or device, and loading and executing the program code stored by the system or device computer (or CPU 3 or MPU) on the recording medium.

[0099]

In that case, the program code itself that is loaded from the recording medium achieves the novel functions of the present invention, and the recording medium storing that program code constructs the present invention.

[0100]

The recording medium for supplying the program code may for example include a floppy disk, a hard disk, an optical disk, an optomagnetic disk, a CD-ROM, a CD-R, magnetic tape, a nonvolatile memory card, or the ROM 4, etc. The program code may also be supplied from a server computer by way of other MIDI devices or a communication network.

[0101]

Needless to say, executing the program code read out from the computer not only implements the functions of the above described embodiments but also includes the case where all or a portion of that actual processing is implemented by an OS running on a computer, based on the instructions in that program code to execute the functions of the above described embodiments.

[0102]

Moreover, needless to say, the present invention also includes the case where, after the program code read out from the recording medium is written into the memory including a function expansion board inserted into the computer or a function expansion unit connected to a computer; the CPU 3 contained in that function expansion board or function expansion unit executes all or a portion of the actual processing; and by means of that processing executes the functions of the above described embodiments.

[0103]

[Effect of the Invention]

As previously described for the invention according to claim 1 or claim 5, when waveform data relating to a readout instruction is stored in the second storage means, then music sounds are generated based on that stored waveform data. When there is no waveform data relating to a readout instruction stored in the second storage means, then that waveform data is read out from the first storage means and after being transferred to the second storage means, then music sounds

are generated based on that stored waveform data. In other words, when generating music sounds, the waveform data is supplied at high speed using only waveform data stored in the second storage means possessing a high transfer speed and therefore the limit on the number of music sound channels dependent on the first storage means possessing a slow transfer speed can be eliminated.

[0104]

The transfer of waveform data from the first storage means to the second storage means, and the generating of music sounds based on waveform data read out from the second storage means are carried out asynchronously so that there are no restrictions on the method for transferring waveform data to the second storage means, and an efficient transfer methods such as block transfer or burst transfer can be utilized.

[0105]

In the present invention according to the claim 2, the waveform data relating to the readout instruction is transferred to the second storage means in waveform data units so that data management is simple since for example, methods are not utilized that divide the waveform data such that only a portion of the data is transferred to the second storage means.

[0106]

In the present invention according to claim 3 or claim 6, when waveform data with a high priority due to a tone switching instruction for a specified tone is not stored in the second storage means, the waveform data is read out from the first storage means, and transferred to the second storage means so that cache hits for the waveform data used first can be obtained with good efficiency.

[0107]

In the present invention according to claim 4 or claim 7, when the data relating to the music sound generating instruction is not stored in the second storage means, the transfer speed of each waveform sample of the waveform data is read at a higher transfer speed than the waveform sample read out speed during generating of music sounds based on waveform samples that were read from the second storage means.

In this way, the state is eliminated where waveform data to be read out from the second storage means during generation of music sounds is not transferred, and so the present invention can achieve stable generation of music sounds.

[Brief Description of the Drawings]

FIG. 1 is a block diagram showing the overall structure of the music sound generator device of the embodiment of the present invention;

FIG. 2 is a drawing showing the data format of the channel data stored in the register of FIG. 1;

FIG. 3 is a drawing showing the data format of the C control data stored in the register of FIG. 1;

FIG. 4 is a flow chart showing the sequence for tone change event processing performed in particular by the CPU in the music sound generator device of FIG. 1;

FIG. 5 is a flow chart showing the sequence of the note-on event processing executed by the CPU of FIG. 1;

FIG. 6 is a flow chart showing in detail the sequence of the processing subroutine for the load instruction of FIG. 4 or FIG. 5;

FIG. 7 is a flow chart showing the sequence of the transfer processing for executing the cache controller of FIG. 1.

[Explanation of the reference numerals]

3: CPU (decision means, transfer means, tone switching instruction means)

10: waveform memory (first storage means)

11: cache memory (second storage means)

12: cache controller (transfer means, music sound generating means)

13: address converter (transfer means, music sound generating means)

15: DSP (music sound generating means)

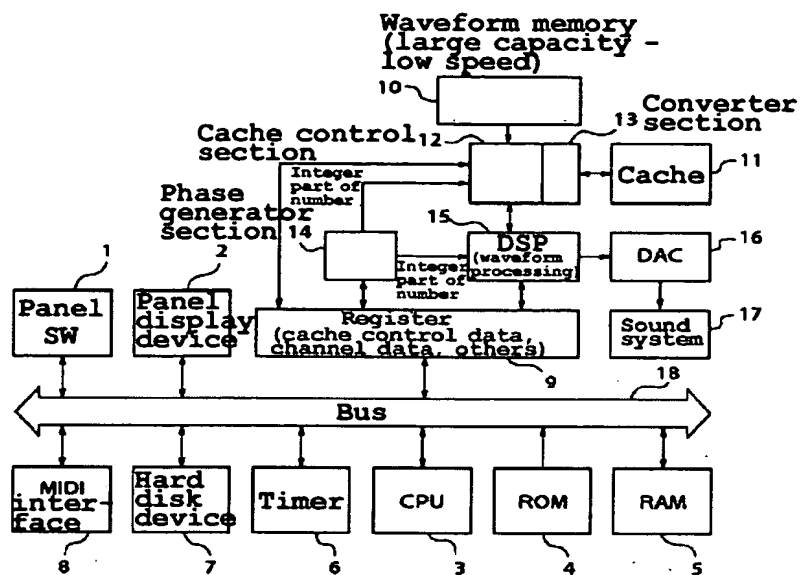


FIG. 1

Channel data

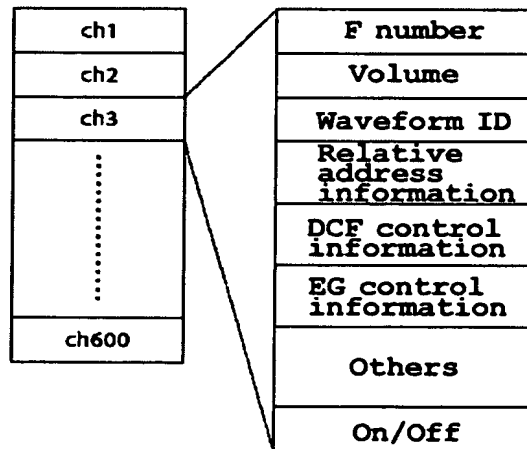


FIG. 2

Cache control data

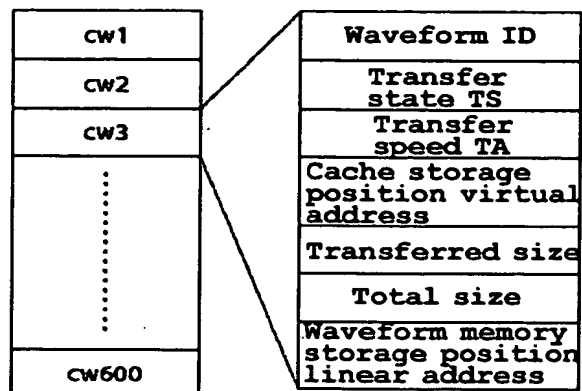


FIG. 3

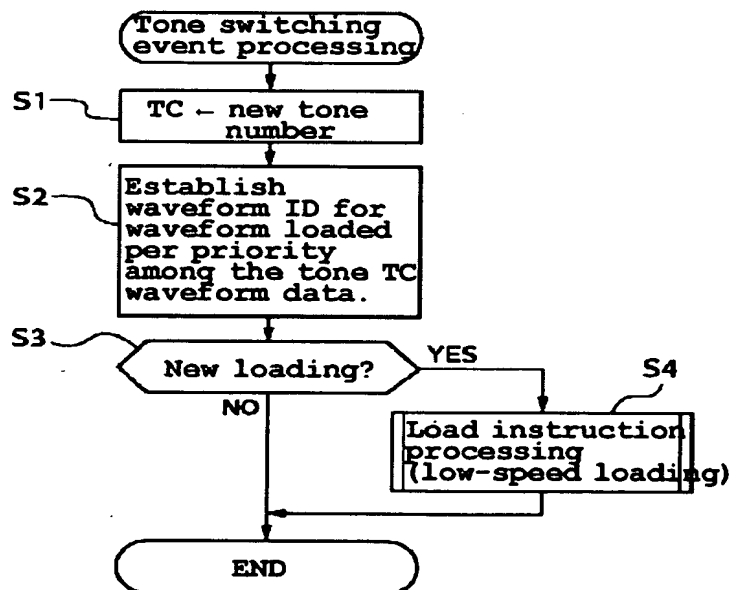


FIG. 4

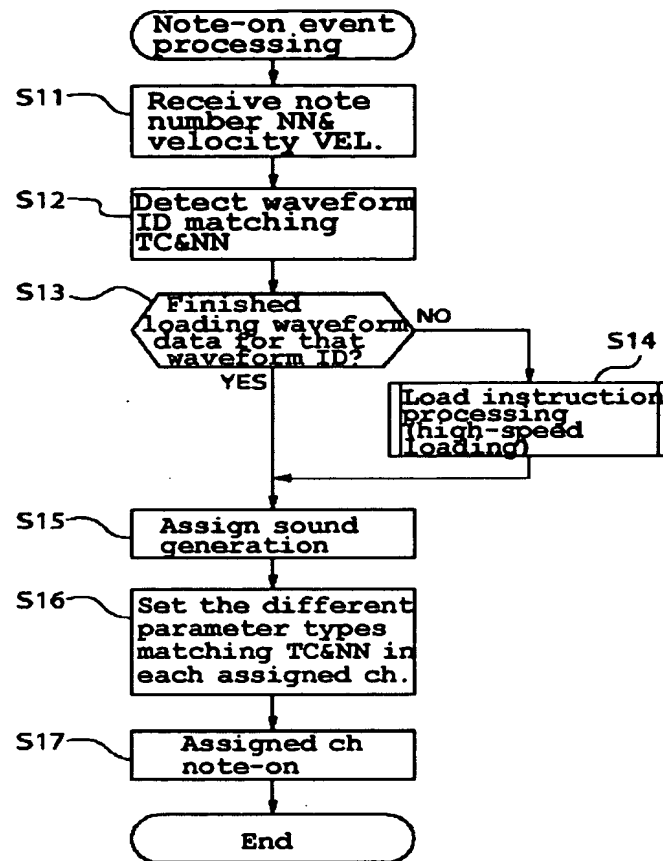


FIG. 5

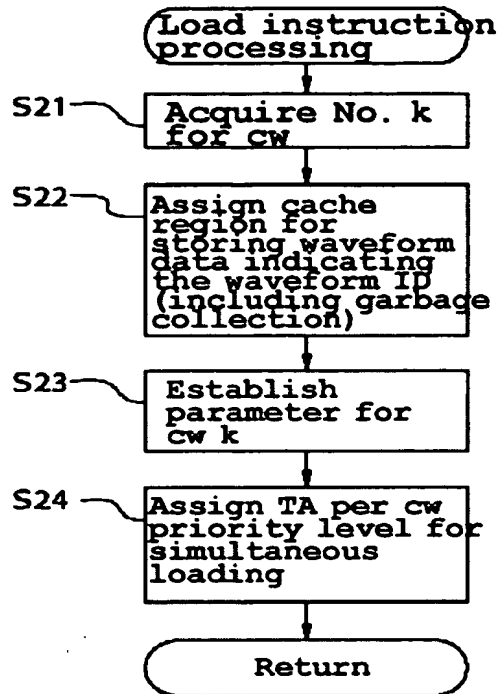


FIG. 6

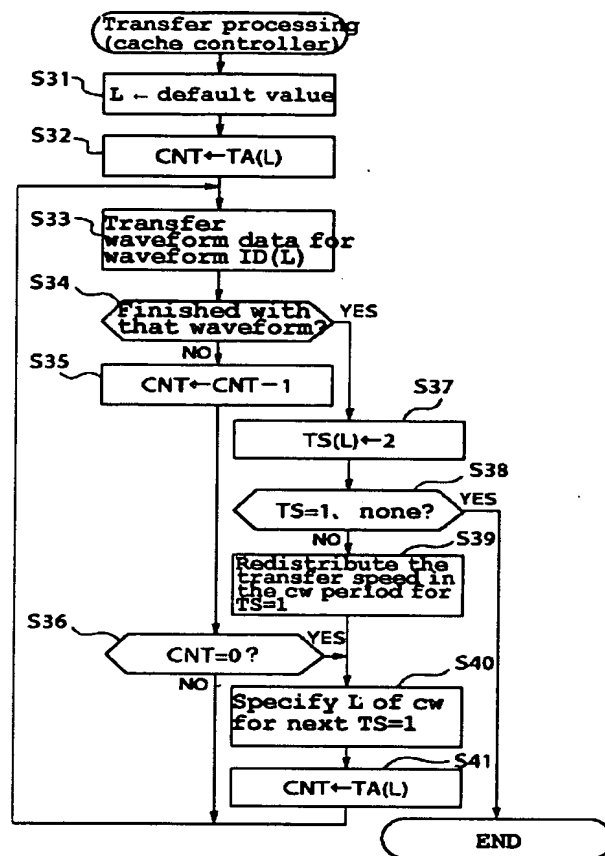


FIG. 7

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開2000-276172

(P2000-276172A)

(43)公開日 平成12年10月6日(2000.10.6)

(51)IntCl.

識別記号

F I

テ-マ-ト*(参考)

G 1 0 H 7/02

G 1 0 H 7/00

5 2 1 K 5 B 0 0 5

G 0 6 F 12/08

G 0 6 F 12/08

U 5 D 3 7 8

審査請求 未請求 請求項の数7 OL (全 13 頁)

(21)出願番号 特願平11-79523

(22)出願日 平成11年3月24日(1999.3.24)

(71)出願人 000004075

ヤマハ株式会社

静岡県浜松市中沢町10番1号

(72)発明者 岡崎 雅嗣

静岡県浜松市中沢町10番1号 ヤマハ株式

会社内

(74)代理人 100081880

弁理士 渡部 敏彦

Fターム(参考) 5B005 JJ11 KK12 LL00 MM01

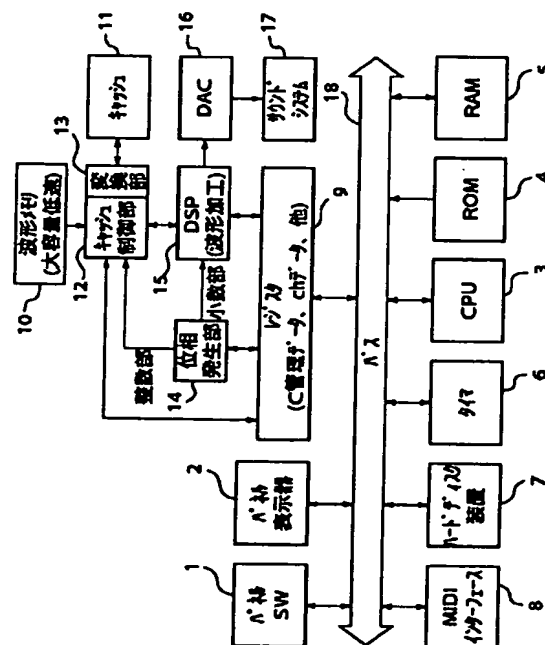
5D378 AD01 AD21 BB11 CC46

(54)【発明の名称】 楽音生成装置および記憶媒体

(57)【要約】

【課題】 データ転送速度が低速の記憶装置に記憶された波形データに基づいて、該記憶装置のデータ転送速度に依存して制限されるチャンネル数より多いチャンネル数の楽音を生成することが可能な楽音生成装置および記憶媒体を提供する。

【解決手段】 発音が指示された波形データのキャッシュヒットが検出されたときには、キャッシュメモリ11上に保持されている波形データを用いて楽音が生成され、発音が指示された波形データのキャッシュミスが検出されたときには、発音指示に係る波形データが波形メモリ10から読み出されて、キャッシュメモリ11に転送され、転送後の波形データを用いて楽音が生成される。また、音色切換が指示されたときに、その切換指示された音色で優先的に使用される波形データが、キャッシュメモリ11上に保持されているか否かが検出され、この波形データがキャッシュメモリ11上に保持されていないときには、この波形データが波形メモリ10から読み出されて、キャッシュメモリ11に転送される。



【特許請求の範囲】

【請求項 1】 複数の波形データを記憶する、データ転送速度が低速の第 1 の記憶手段と、データ転送速度が高速の第 2 の記憶手段と、読み出し指示に係る波形データが前記第 2 の記憶手段に記憶されているか否かを判別する判別手段と、該判別手段による判別の結果、前記読み出し指示に係る波形データが前記第 2 の記憶手段に記憶されていないときには、前記読み出し指示に応じて、該読み出し指示に係る波形データを前記第 1 の記憶手段から読み出し、前記第 2 の記憶手段に転送する転送手段と、前記第 2 の記憶手段からその記憶された波形データを読み出し、該読み出された波形データに基づいて楽音を生成する楽音生成手段とを有し、前記転送手段による波形データの転送と前記楽音生成手段による楽音の生成とは非同期に行われることを特徴とする楽音生成装置。

【請求項 2】 前記転送手段は、前記読み出し指示に係る波形データを波形データ単位で前記第 2 の記憶手段に転送することを特徴とする請求項 1 に記載の楽音生成装置。

【請求項 3】 複数の波形データを記憶する、データ転送速度が低速の第 1 の記憶手段と、データ転送速度が高速の第 2 の記憶手段と、音色切換を指示する音色切換指示手段と、該音色切換指示手段により音色切換が指示されたときに、該指示された音色において優先的に用いられる波形データが前記第 2 の記憶手段に記憶されているか否かを判別する判別手段と、該判別手段による判別の結果、前記優先的に用いられる波形データが前記第 2 の記憶手段に記憶されていないときには、該波形データを前記第 1 の記憶手段から読み出して、前記第 2 の記憶手段に転送する転送手段と、前記第 2 の記憶手段からその記憶された波形データを読み出し、該読み出された波形データに基づいて楽音を生成する楽音生成手段とを有することを特徴とする楽音生成装置。

【請求項 4】 複数の波形サンプルからなる波形データを複数個記憶する、データ転送速度が低速の第 1 の記憶手段と、データ転送速度が高速の第 2 の記憶手段と、発音指示に係る波形データが前記第 2 の記憶手段に記憶されているか否かを判別する判別手段と、該判別手段による判別の結果、前記発音指示に係る波形データが前記第 2 の記憶手段に記憶されていないときには、前記発音指示に応じて、該発音指示に係る波形データの各波形サンプルを前記第 1 の記憶手段から読み出し、前記第 2 の記憶手段に転送する転送手段と、前記発音指示に応じて、前記第 2 の記憶手段から該発音指示に係る波形データの波形サンプルを順次読み出し、

該読み出された波形サンプルに基づいて楽音を生成する楽音生成手段とを有し、

前記転送手段による波形データの各波形サンプルの転送速度を、前記楽音生成手段による各波形サンプルの読み出し速度より高速にしたことを特徴とする楽音生成装置。

【請求項 5】 読み出し指示に係る波形データが、データ転送速度が高速の第 2 の記憶手段に記憶されているか否かを判別する判別モジュールと、

該判別モジュールによる判別の結果、前記読み出し指示に係る波形データが前記第 2 の記憶手段に記憶されていないときには、前記読み出し指示に応じて、該読み出し指示に係る波形データを、複数の波形データが記憶されている、データ転送速度が低速の第 1 の記憶手段から読み出し、前記第 2 の記憶手段に転送する転送モジュールと、

前記第 2 の記憶手段からその記憶された波形データを読み出し、該読み出された波形データに基づいて楽音を生成する楽音生成モジュールとを含み、

前記転送モジュールによる波形データの転送と前記楽音生成モジュールによる楽音の生成とは非同期に行われることを特徴とする、コンピュータが実現できるプログラムを格納した記憶媒体。

【請求項 6】 音色切換を指示する音色切換指示モジュールと、

該音色切換指示モジュールにより音色切換が指示されたときに、該指示された音色において優先的に用いられる波形データが、データ転送速度が高速の第 2 の記憶手段に記憶されているか否かを判別する判別モジュールと、

該判別モジュールによる判別の結果、前記優先的に用いられる波形データが前記第 2 の記憶手段に記憶されていないときには、該波形データを、複数の波形データが記憶されている、データ転送速度が低速の第 1 の記憶手段から読み出して、前記第 2 の記憶手段に転送する転送モジュールと、

前記第 2 の記憶手段からその記憶された波形データを読み出し、該読み出された波形データに基づいて楽音を生成する楽音生成モジュールとを含む、コンピュータが実現できるプログラムを格納した記憶媒体。

【請求項 7】 発音指示に係る波形データが、データ転送速度が高速の第 2 の記憶手段に記憶されているか否かを判別する判別モジュールと、

該判別モジュールによる判別の結果、前記発音指示に係る波形データが前記第 2 の記憶手段に記憶されていないときには、前記発音指示に応じて、該発音指示に係る波形データの各波形サンプルを、複数の波形サンプルからなる波形データが複数個記憶されている、データ転送速度が低速の第 1 の記憶手段から読み出し、前記第 2 の記憶手段に転送する転送モジュールと、

前記発音指示に応じて、前記第 2 の記憶手段から該発音

指示に係る波形データの波形サンプルを順次読み出し、該読み出された波形サンプルに基づいて楽音を生成する楽音生成モジュールとを含み、前記転送モジュールによる波形データの各波形サンプルの転送速度を、前記楽音生成モジュールによる各波形サンプルの読み出し速度より高速にしたことを特徴とする、コンピュータが実現できるプログラムを格納した記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データ転送速度が低速の記憶装置に記憶された波形データに基づいて、該記憶装置のデータ転送速度に依存して制限されるチャンネル数より多いチャンネル数の楽音を生成する楽音生成装置および記憶媒体に関する。

【0002】

【従来の技術】従来、データ転送速度が低速の記憶装置（波形メモリ（ROM）を含む）に記憶された波形データに基づいて楽音を生成する楽音生成装置として、次のものが知られている。すなわち、

（1）波形メモリから読み出された波形サンプルを補間することにより、目的の楽音波形を生成する楽音生成装置において、補間に用いる読み出し波形サンプルをキャッシュメモリに記憶し、このキャッシングされた波形サンプルに基づいて楽音波形を生成するもの

（2）データ転送速度が低速の記憶装置であるハードディスクに記憶された波形データを、低容量かつデータ転送速度が高速のRAMにロードし、このロードされた波形データに基づいて楽音を生成する楽音生成装置等である。

【0003】

【発明が解決しようとする課題】しかし、上記従来の楽音生成装置のうち（1）では、生成すべき楽音波形が進行し、この楽音波形を補間生成するための波形サンプルがキャッシュメモリに保持されていないときには、その波形サンプルを波形メモリから読み出さなければならぬので、結局、生成される楽音波形のチャンネル数は波形メモリのデータ転送速度に依存することになる。

【0004】また、上記従来の楽音生成装置のうち

（2）では、発音が指示されたときに、その発音指示された楽音に対応する波形データがRAM上にロードされていない場合には、その楽音を生成できなかった。

【0005】本発明は、この点に着目してなされたものであり、データ転送速度が低速の記憶装置に記憶された波形データに基づいて、該記憶装置のデータ転送速度に依存して制限されるチャンネル数より多いチャンネル数の楽音を生成することが可能な楽音生成装置および記憶媒体を提供することを目的とする。

【0006】

【課題を解決するための手段】上記目的を達成するた

め、請求項1に記載の楽音生成装置は、複数の波形データを記憶する、データ転送速度が低速の第1の記憶手段と、データ転送速度が高速の第2の記憶手段と、読み出し指示に係る波形データが前記第2の記憶手段に記憶されているか否かを判別する判別手段と、該判別手段による判別の結果、前記読み出し指示に係る波形データが前記第2の記憶手段に記憶されていないときには、前記読み出し指示に応じて、該読み出し指示に係る波形データを前記第1の記憶手段から読み出し、前記第2の記憶手段に転送する転送手段と、前記第2の記憶手段からその記憶された波形データを読み出し、該読み出された波形データに基づいて楽音を生成する楽音生成手段とを有し、前記転送手段による波形データの転送と前記楽音生成手段による楽音の生成とは非同期に行われることを特徴とする。

【0007】ここで、第1の記憶手段は、典型的には、波形メモリ（ROM）であるが、これに限らず、ハードディスクやMD等、データ転送速度が低速の記憶手段であればどのようなものであってもよい。第2の記憶手段は、典型的には、キャッシュメモリであるが、これに限らず、データ転送速度が高速であれば、RAMであってもよい。以上は、請求項が変わっても同様である。

【0008】好ましくは、前記転送手段は、前記読み出し指示に係る波形データを波形データ単位で前記第2の記憶手段に転送することを特徴とする。

【0009】請求項3に記載の楽音生成装置は、複数の波形データを記憶する、データ転送速度が低速の第1の記憶手段と、データ転送速度が高速の第2の記憶手段と、音色切換を指示する音色切換指示手段と、該音色切換指示手段により音色切換が指示されたときに、該指示された音色において優先的に用いられる波形データが前記第2の記憶手段に記憶されているか否かを判別する判別手段と、該判別手段による判別の結果、前記優先的に用いられる波形データが前記第2の記憶手段に記憶されていないときには、該波形データを前記第1の記憶手段から読み出して、前記第2の記憶手段に転送する転送手段と、前記第2の記憶手段からその記憶された波形データを読み出し、該読み出された波形データに基づいて楽音を生成する楽音生成手段とを有することを特徴とする。

【0010】請求項4に記載の楽音生成装置は、複数の波形サンプルからなる波形データを複数個記憶する、データ転送速度が低速の第1の記憶手段と、データ転送速度が高速の第2の記憶手段と、発音指示に係る波形データが前記第2の記憶手段に記憶されているか否かを判別する判別手段と、該判別手段による判別の結果、前記発音指示に係る波形データが前記第2の記憶手段に記憶されていないときには、前記発音指示に応じて、該発音指示に係る波形データの各波形サンプルを前記第1の記憶手段から読み出し、前記第2の記憶手段に転送する転送

手段と、前記発音指示に応じて、前記第2の記憶手段から該発音指示に係る波形データの波形サンプルを順次読み出し、該読み出された波形サンプルに基づいて楽音を生成する楽音生成手段とを有し、前記転送手段による波形データの各波形サンプルの転送速度を、前記楽音生成手段による各波形サンプルの読み出し速度より高速にしたことを特徴とする。

【0011】また、上記目的を達成するために、請求項5に記載の記憶媒体は、読み出し指示に係る波形データが、データ転送速度が高速の第2の記憶手段に記憶されているか否かを判別する判別モジュールと、該判別モジュールによる判別の結果、前記読み出し指示に係る波形データが前記第2の記憶手段に記憶されていないときには、前記読み出し指示に応じて、該読み出し指示に係る波形データを、複数の波形データが記憶されている、データ転送速度が低速の第1の記憶手段から読み出し、前記第2の記憶手段に転送する転送モジュールと、前記第2の記憶手段からその記憶された波形データを読み出し、該読み出された波形データに基づいて楽音を生成する楽音生成モジュールとを含み、前記転送モジュールによる波形データの転送と前記楽音生成モジュールによる楽音の生成とは非同期に行われることを特徴とする。

【0012】請求項6に記載の記憶媒体は、音色切換を指示する音色切換指示モジュールと、該音色切換指示モジュールにより音色切換が指示されたときに、該指示された音色において優先的に用いられる波形データが、データ転送速度が高速の第2の記憶手段に記憶されているか否かを判別する判別モジュールと、該判別モジュールによる判別の結果、前記優先的に用いられる波形データが前記第2の記憶手段に記憶されていないときには、該波形データを、複数の波形データが記憶されている、データ転送速度が低速の第1の記憶手段から読み出して、前記第2の記憶手段に転送する転送モジュールと、前記第2の記憶手段からその記憶された波形データを読み出し、該読み出された波形データに基づいて楽音を生成する楽音生成モジュールとを含むことを特徴とする。

【0013】請求項7に記載の記憶媒体は、発音指示に係る波形データが、データ転送速度が高速の第2の記憶手段に記憶されているか否かを判別する判別モジュールと、該判別モジュールによる判別の結果、前記発音指示に係る波形データが前記第2の記憶手段に記憶されていないときには、前記発音指示に応じて、該発音指示に係る波形データの各波形サンプルを、複数の波形サンプルからなる波形データが複数個記憶されている、データ転送速度が低速の第1の記憶手段から読み出し、前記第2の記憶手段に転送する転送モジュールと、前記発音指示に応じて、前記第2の記憶手段から該発音指示に係る波形データの波形サンプルを順次読み出し、該読み出された波形サンプルに基づいて楽音を生成する楽音生成モジュールとを含み、前記転送モジュールによる波形データ

の各波形サンプルの転送速度を、前記楽音生成モジュールによる各波形サンプルの読み出し速度より高速にしたことを特徴とする。

【0014】

【発明の実施の形態】以下、本発明の実施の形態を図面に基づいて詳細に説明する。

【0015】図1は、本発明の一実施の形態に係る楽音生成装置の概略構成を示すブロック図である。

【0016】同図に示すように、本実施の形態の楽音生成装置は、各種情報を入力するための複数のスイッチを備えたパネルスイッチ1と、各種情報等を表示する、たとえば大型液晶ディスプレイ(LCD)および発光ダイオード(LED)等を備えた表示装置2と、装置全体の制御を司るCPU3と、該CPU3が実行する制御プログラムや各種テーブルデータ等を記憶するROM4と、演奏データ、各種入力情報および演算結果等を一時的に記憶するRAM5と、タイマ割込み処理における割込み時間や各種時間を計時するタイマ6と、前記制御プログラムを含む各種アプリケーションプログラムや各種データ等を記憶するハードディスク装置7と、外部からのMIDI (musical instrument digital interface) 信号を入力したり、MIDI 信号を外部に出力したりするMIDI インターフェース8と、ch (channel) データ(図2を用いて後述)やC (cache) 管理データ(図3を用いて後述)等のデータを格納する領域を備えたレジスタ9とを有し、これらの各構成要素1~9は、バス18を介して相互に接続されている。

【0017】ハードディスク装置7には、前述のように、CPU3が実行する制御プログラムも記憶でき、ROM4に制御プログラムが記憶されていない場合には、このハードディスクに制御プログラムを記憶させておき、それをRAM5に読み込むことにより、ROM4に制御プログラムを記憶している場合と同様の動作をCPU3にさせることができる。このようにすると、制御プログラムの追加やバージョンアップ等が容易に行える。

【0018】MIDI インターフェース8は、専用のものに限らず、RS-232CやUSB (ユニバーサル・シリアル・バス)、IEEE1394 (アイトリプルイー1394)等の汎用のインターフェースより構成してもよい。この場合、MIDI メッセージ以外のデータをも同時に送受信してもよい。

【0019】レジスタ9には、大容量(たとえば、128Mword)かつデータ転送速度が低速(たとえば、6Mword/秒)の波形メモリ(ROM)10に記憶されている波形データを、小容量(たとえば、4Mword)かつデータ転送速度が高速(たとえば、66Mword/秒)のキャッシュメモリ11にロードしたり、キャッシュメモリ11にロードされている波形データを読み出したりするキャッシュ制御部12と、波形データを読み出す際の1サンプル当たりのアドレスの進み量を

示すFナンバF Nを各発音チャンネル毎に累算することにより、該各チャンネル毎の波形データの読み出しアドレス（相対アドレス）を発生する位相発生部14と、キャッシュメモリ11から読み出された波形データを加工するためのDSP（digital signal processor）15が接続されている。

【0020】ここで、本実施の形態の楽音生成装置は、楽音を生成するときには、常にキャッシュメモリ11からその楽音生成の基準となる波形データを読み出し、この波形データに基づいて楽音を生成する。そして、その楽音生成の基準となる波形データがキャッシュメモリ11上に保持されていないとき、すなわちキャッシュミスが発生したときには、その波形データを波形メモリ10から読み出して、キャッシュメモリ11に転送した後、この波形データをキャッシュメモリ11から読み出して楽音を生成する。このとき、後述するように、波形メモリ10からキャッシュメモリ11への波形データの転送速度は、キャッシュメモリ11上の波形データの読み出しアドレスの進み量（読み出し速度）より速く設定されているので、発音の途中で、キャッシュメモリ11に読み出すべき波形サンプルが未転送という状態にはならない。

【0021】本実施の形態では、装置のサンプリング周波数を、たとえば50kHzとし、位相発生部14によって発生された読み出しアドレスに小数部が含まれるときには、そのアドレスの波形サンプルは直線補間によって算出するようにしているため、1周期当たり2個の波形サンプルを読み出す必要があるため、サンプリング周波数は実質100kHzとなっている。

【0022】この100kHzのサンプリング周波数の1周期で、1個の波形サンプル（そのデータ容量は1wordとする）を読み出せば、1チャンネル分の楽音を生成できるので、本実施の形態のように、常にキャッシュメモリ11をアクセスすることにより、その記憶された波形データ（波形サンプル）を読み出して楽音生成に使用したときには、 $66\text{Mword}/\text{秒} \div 100\text{kwor d}/\text{秒}/\text{ch} = 660\text{ch}$ 分の楽音を生成することができる。

【0023】ただし、本実施の形態では、1サンプリング周期内で、最大6Mword/秒のデータ転送、すなわち波形メモリ10からキャッシュメモリ11への波形データのデータ転送も行っているため、 $6\text{Mword}/\text{秒} \div 100\text{kwor d}/\text{秒}/\text{ch} = 60\text{ch}$ 分、上記660chから差し引く必要がある。したがって、本実施の形態の楽音生成装置は、最大600chの楽音を生成することができる。

【0024】位相発生部14は、レジスタ9に格納された各発音チャンネル（チャンネル数は、上述のように、最大600ch）毎のchデータに基づいて、該各チャンネル毎の波形データの読み出しアドレス（実数値）を

発生し、その整数部をキャッシュ制御部12に供給するとともに、その小数部をDSP15に供給する。

【0025】キャッシュ制御部12は、レジスタ9に格納されたC管理データに基づいて、各チャンネルで読み出そうとしている波形データがキャッシュメモリ11に記憶されているか否かを判別し、記憶されていないときには、当該チャンネルにおける波形データの読み出しを開始すると同時に、その波形データを波形メモリ10から読み出してキャッシュメモリ11に転送する一方、記憶されているときには、当該チャンネルで発生された読み出しアドレスの整数部に対応する位置（およびその次の位置）の波形データをキャッシュメモリ11から読み出して、DSP15に供給する。

【0026】ここで、波形データは、波形メモリ10からブロック単位（1ブロックは、本実施の形態では、1keyword）で読み出されて、キャッシュメモリ11にロードされる。通常、読み出すべき波形データの容量は1keywordより大きいので、複数ブロックがキャッシュメモリ11にロードされるが、この複数ブロック分の容量のまとまった空き領域がキャッシュメモリ11に存在しない場合には、各ブロックの波形データは、キャッシュメモリ11上で離散的に記憶される。他方、位相発生部14は、読み出しアドレスを、当該波形データのアドレスが連続しているものとして発生する。このため、キャッシュメモリ11上の実アドレスに対して仮想アドレスを定義し、この仮想アドレス（各ブロック毎の仮想アドレス）を再配置することにより、実アドレスでは依然として離散的に存在している波形データのアドレスを、見かけ上連続的にしている。この種の技術を、ガーベジ・コレクション（garbage collection）と呼んでいる。これにより、空き領域をまとめる際にキャッシュメモリ11上に記憶されているデータを転送する必要がなくなるので、処理を高速に行うことができる。

【0027】キャッシュ制御部12は、目的の波形データをキャッシュメモリ11から読み出すために、この仮想アドレスをキャッシュメモリ11上の実アドレスに変換するアドレス変換部13を介して、キャッシュメモリ11に接続されている。

【0028】DSP15は、発音チャンネル処理、エフェクト処理およびミキサ処理を行い、これら各処理の施されたデジタル楽音データをDAC（digital to analog converter）16に出力し、DAC16は、DSP15から供給されたデジタル楽音データをアナログ楽音信号に変換し、アンプやスピーカ等からなるサウンドシステム17を介して音響に変換される。

【0029】ここで、発音チャンネル処理の主なものとしては、たとえば、サンプル間補間処理、音色フィルタ処理、音量エンベロープ処理、チャンネル累算処理が挙げられる。

【0030】サンプル間補間処理では、キャッシュ制御

部12から供給された波形データと位相発生部14から供給された読み出しアドレスの小数部に基いて補間演算を行い、補間された波形データの波形サンプルを生成する。本実施の形態では、キャッシュ制御部12は、各チャンネル当たり2個の波形サンプルをDSP15に供給し、DSP15は、この2個の波形サンプルに基づいて直線補間により補間サンプルを生成する。もちろん、過去に読み出された波形サンプルを記憶しておき、この波形サンプルと新たに読み出された波形サンプルに基づいて、より高次の補間を行うようにしてもよい。

【0031】音色フィルタ処理では、上記補間後の波形データに対して、たとえばローパスフィルタ演算により音色制御を行う。このローパスフィルタ演算は、パラメータとして供給されたカットオフ特性やレゾナンス特性等に基づいて行われる。

【0032】音量エンベロープ処理では、音色フィルタ処理が施された波形データに対して、楽音の立ち上がりから立ち下がりまでの音量の時間変化（エンベロープ）を与える。この音量エンベロープ処理も、パラメータとして供給されたEG（envelope generator）制御情報に

基づいて行われる。

【0033】チャンネル累算処理では、エンベロープの付与された波形データを、各エフェクト処理（本実施の形態では、コーラス処理、リバーブ処理およびイコライザ処理の3種類の処理がある）向けにそれぞれレベル制御した後、全チャンネル分累算し、各エフェクト処理に出力する。

【0034】コーラス処理では、当処理向けに出力された波形データに対してコーラス効果を付与する。

【0035】リバーブ処理では、当処理向けに出力された波形データに対して残響効果を付与する。

【0036】ミキサ処理では、コーラス効果が付与された波形データと残響効果が付与された波形データの各音量を制御し、この音量制御された各波形データとイコライザ処理向けに出力された波形データとをミキシングする。

【0037】イコライザ処理では、ミキサ処理によってミキシングされた波形データの周波数特性を加工制御する。

【0038】この周波数特性が加工制御された波形データが、DSP15からDAC16に供給される。

【0039】以上のように、位相発生部14およびDSP15は、それぞれ、1サンプリング周期内で最大600ch分の楽器生成のための処理を時分割で行う必要がある。この処理を実現するためには、必要に応じて、各回路を並列化したり、パイプライン化したりすればよい。すなわち、各処理を時分割600chで複数段にパイプライン化した構成、2つに並列化してそれぞれ時分割300chで複数段にパイプライン化した構成、4つに並列化してそれぞれ時分割150chで複数段にパイ

プライン化した構成等を探ることができる。一般的に、並列化の数を増やすことによりパイプラインの段数を減少させることができる。

【0040】図2は、レジスタ9に格納されるchデータのデータフォーマットを示す図である。

【0041】各chデータは、それぞれ、当該発音チャンネルでの楽音生成のために必要な複数個のデータからなり、レジスタ9には、600ch分のデータが記憶される領域ch1～ch600が設けられている。なお、各chデータのデータフォーマットはすべて共通であるため、同図には、ch3のchデータのデータフォーマットが代表的に例示されている。

【0042】同図において、chデータは、FナンバFNと、生成される楽音の全体的な音量と、楽音生成に使用する、波形メモリ10上の波形データを特定するための波形IDと、波形メモリ10（またはキャッシュメモリ11）上の当該（波形IDに対応する）波形データの先頭記憶位置（アドレス）を基準とした所定位置（たとえば、読み出し開始位置、ループ読み出し開始位置およびループ読み出し終了位置）の相対アドレス情報と、前記音色フィルタ処理でのフィルタ演算に用いるパラメータであるDCF（digital controlled filter）制御情報（前記カットオフ特性やレゾナンス特性を含む）と、前記音量エンベロープ処理での音量制御に用いるパラメータであるEG制御情報と、楽音生成に用いるその他の情報と、ノートオン/オフの状態をそれぞれ“0”と“1”で示すオン/オフデータとによって構成されている。

【0043】FナンバFNは、上述のように、波形データを読み出す際の1サンプル当たりのアドレスの進み量を示すものであり、生成すべき楽音の音高とその生成に用いる波形データが決まれば、その値は決まる。前記位相発生部14は、システムのサンプリング周波数（本実施の形態では、前述のように、50kHz）に同期して、各チャンネル毎にFナンバFNを累積し、該各チャンネル毎の波形データの読み出しアドレス（相対アドレス）を発生する。

【0044】波形メモリ10に記憶されている各波形データには、それぞれIDが付与され、chデータ内の波形IDを指定することにより、目的の波形データを波形メモリ10上で特定することができる。

【0045】オン/オフデータは、その値が“0”から“1”に変化した時点で、当該チャンネルの楽音の生成を開始し、“1”から“0”に変化した時点で、当該チャンネルの楽音のリリースを開始するように、前記DSP15に指示するためのものである。

【0046】図3は、レジスタ9に格納されるC管理データのデータフォーマットを示す図である。

【0047】C管理データは、キャッシュメモリ11に格納される波形データの状態を管理するために必要な複

数個のデータからなり、レジスタ9には、600波形分のデータが記憶される領域cw1～cw600が設けられている。なお、各C管理データのデータフォーマットはすべて共通であるため、同図には、cw (cache wave) 3のC管理データのデータフォーマットが例示されている。

【0048】同図において、C管理データは、上記chデータ内の波形IDと同様に作用する波形IDと、キャッシュメモリ11への当該波形データの転送状態を示す転送ステートTSと、キャッシュメモリ11へ当該波形データを転送するときの速度を示す転送速度TAと、当該波形データのキャッシュメモリ11上の記憶位置を仮想アドレスで表現するキャッシュ記憶位置仮想アドレスと、当該波形データのうちキャッシュメモリ11に転送されたデータのデータサイズを示す転送済みサイズと、当該波形データの全データサイズを示すトータルサイズと、当該波形データの波形メモリ10上の記憶位置をリアルアドレス（波形データは通常波形メモリ10上リニアに（連続して）記憶されているため、リニアアドレスである）で示す波形メモリ記憶位置リニアアドレスとによって構成されている。

【0049】転送ステートTSは、“0”から“2”までのいずれかの整数値を採り、各整数値は、次のような状態を意味している。

【0050】0：当該波形IDが示す波形データは楽音生成に使用されていない状態

1：当該波形IDが示す波形データはキャッシュメモリ11に転送中の状態

2：当該波形IDが示す波形データはキャッシュメモリ11に転送を完了した状態

転送速度TAは、“0”から“120”までのいずれかの整数値を採り、1秒あたりに転送される前記ブロックのブロック数（本実施の形態では、50ブロック＝50kword）を1単位としている。たとえば、TA＝0は、キャッシュメモリ11へのデータ転送を行わない状態を示し、TA＝1は、キャッシュメモリ11への50ブロック/秒のデータ転送を行う状態を示し、TA＝2は、キャッシュメモリ11への100ブロック/秒のデータ転送を行う状態を示し、というように、転送速度TAとして設定される値が大きくなるほど、1秒間に転送されるブロック数、すなわち転送速度が速くなるようになっている。

【0051】ここで、1秒あたりに転送されるブロック数の1単位を50ブロックとしたのは、この転送速度が、FナンバFNが“1”に設定されたとき（システムのサンプリング周波数（＝50kHz）で波形データの各波形サンプルを読み出すとき）の楽音再生に相当する速度であるからである。

【0052】また、転送速度TAとしては、上述のように、最大“120”を採ることができる。これは、波形

メモリ10からのデータ転送速度が、本実施の形態では6Mword/秒であることによる。したがって、キャッシュメモリ11に転送すべき波形データが複数個あるときには、“120”を分配する必要がある。（すなわち、各ロードに対して割り当てた転送速度TAの合計値は“120”を超えない。）分配する方法としては、たとえば次のような方法が考えられる。すなわち、

（1）発音時のロードに対して、音色切替時のロードより、大きい整数値を優先的に割り当てる

（2）発音時のロードに対しては、その波形データを使用する発音チャンネルのFナンバFNに応じた整数値以上の数を割り当てる

（3）音色切替時のロードに対しては、120以下の任意の整数値を割り当ててもよい（たとえば、他のロードに割り当てた転送速度TAの合計が、“120”に到達してしまっただけの場合には、残りの音色切替時のロードに対しては、転送速度TAを“0”としてもよい）

ここで、上記（1）において、発音時のロードとは、今まさに発音されようとしている楽音に対応する波形データがキャッシュメモリ11上に記憶されていないときに、当該波形データを波形メモリ10から読み出してキャッシュメモリ11にロードすることをいう。また、音色切替時のロードとは、たとえば演奏データとして音色切替を指示するデータが供給されたときに、その指示された音色で優先的に使用される（たとえば、最も頻繁に使用される）波形データがキャッシュメモリ11上に記憶されていない場合には、本実施の形態では、当該波形データを波形メモリ10から読み出してキャッシュメモリ11にロードするが、このロードをいう。

【0053】また、上記（2）において、FナンバFNに応じた整数値とは、たとえば、FナンバFNの小数点を切り上げた値をいい、FN＝1.5のときには“2”である。転送速度TAとして、FナンバFNに応じた整数値以上の整数値を設定するようにしたのは、転送速度の1単位を、FナンバFN＝1のときに必要な転送速度を基準に設定したからであり、FナンバFNが“1”より大きい値に設定されている場合には、それに見合った転送速度で波形データをキャッシュメモリ11に転送しないと、読み出しアドレスの進み量が転送済みの波形サンプルのアドレスを越えてしまうことがあり、このときには、楽音を再生できなくなってしまうからである。

【0054】さらに、上記（3）において、音色切替時のロードに対する転送速度TAの設定値に、上記（2）のような条件を設けなかったのは、音色切替の指示からその音色で実際に楽音が生成されるまでの間に時間的な余裕があるからである。

【0055】キャッシュ記憶位置仮想アドレスは、上述のように、当該波形データのキャッシュメモリ11上の記憶位置を仮想アドレスで表現したものであり、キャッシュメモリ11の実アドレス空間上で離散的に配置され

ている、当該波形データを構成する各ブロックを、仮想アドレス空間上で連続的に再配置したものである。本実施の形態では、このキャッシュ記憶位置仮想アドレスに基づいて、当該波形データの各波形サンプルを読み出すようにしているので、位相発生部14は、単にFナンバFNを累算するだけで読み出しアドレスを発生させることができる。仮想アドレス空間上でのデータブロックの再配置は、前記キャッシュ制御部12が自動的にを行い、この結果に基づいて、キャッシュ制御部12は、仮想アドレスと実アドレスとを相互に変換するためにアドレス変換部13内に設けられたアドレス変換テーブルを書き換える。このキャッシュ制御部12が実行する制御処理については、特開平6-35473号公報に詳細に示されているので、その説明を省略する。この公報に記載されたハードウェア的なアドレス変換を採用することにより、仮想アドレスによるキャッシュメモリ11へのアクセスを時間遅れなく実行することができる。

【0056】以上のように構成された楽音生成装置が実行する制御処理を、まずその概要を説明した後に、図4～図7を参照して詳細に説明する。

【0057】本実施の形態の楽音生成装置は、主として次の制御処理を行う。すなわち、

(1) 発音が指示された波形データのキャッシュヒットまたはキャッシュミスを検出し、キャッシュヒットが検出されたときには、キャッシュメモリ11上に保持されている波形データを用いて楽音を生成し、キャッシュミスが検出されたときには、発音指示に係る波形データを波形メモリ10から読み出して、キャッシュメモリ11に転送し、転送後の波形データを用いて楽音を生成する

(2) 音色切替が指示されたときに、その切替指示された音色で優先的に使用される波形データが、キャッシュメモリ11上に保持されているか否かを検出し、この波形データがキャッシュメモリ11上に保持されていないときには、この波形データを波形メモリ10から読み出して、キャッシュメモリ11に転送する

(3) 上記(1)および(2)における波形メモリ10からキャッシュメモリ11への波形データの転送と、各チャンネルの楽音生成をするための、キャッシュメモリ11に保持されている波形データ(波形サンプル)の読み出しとを非同期に行う

(4) 波形データのキャッシュメモリ11へのキャッシングを1波形単位で行う(ただし、波形メモリ10からの波形データの転送は、1波形単位で行う必要はなく、本実施の形態では、複数に分割された波形を時分割で並行して転送している。要するに、キャッシュヒットの判定や、ヒットしなかった場合のロード指示等のキャッシュ管理が、1つの波形の中の各サンプルあるいは複数サンプル単位ではなく、各1つの波形を単位として行われるということである。また、キャッシュメモリ11上の波形データの格納態様は、離散的であってもよいし、連

続的であってもよい。)

(5) 上記(1)におけるキャッシュミス時の転送、すなわち波形メモリ10からキャッシュメモリ11への波形データの転送を、楽音生成時の波形サンプルの読み出し速度(アドレスの進み量)より速く行う

次に、上記各制御処理(1)～(5)の詳細を説明する。

【0058】図4は、本実施の形態の楽音生成装置、特に前記CPU3が実行する音色切替イベント処理の手順を示すフローチャートであり、上記制御処理(2)を実現したものである。本処理は、演奏データとして音色切替イベントが供給されたときに、起動される。

【0059】同図において、まず、音色切替イベントが指示する新たな音色の音色番号(新音色番号)を取得して、RAM5の所定位置に確保された領域TC(以下、この領域の内容を「音色TC」という)に格納する(ステップS1)。

【0060】次に、音色TCにおいて使用される各種波形データ中、優先度に応じてキャッシュメモリ11にロードする波形データの波形IDを決定する(ステップS2)。

【0061】そして、この決定された波形IDに対応する波形データがキャッシュメモリ11に新規にロードすべきものであるか否かを判別する(ステップS3)。ここで、新規にロードすべきものであるか否かは、次のようにして判別する。

【0062】すなわち、ステップS2で決定された波形IDと同一の波形IDが、前記転送ステートTSが“0”以外のC管理データ中にあるか否かをチェックし、ないときに、その波形IDに対応する波形データを新規にロードすべきものと判別する。

【0063】ステップS3で、目的の波形データは、キャッシュメモリ11に転送中または転送が完了している、新規にロードすべき波形データがないときには、直ちに本音色切替イベント処理を終了する一方、新規にロードすべき波形データがあるときには、その波形データのキャッシュメモリ11へのロード指示を行うロード指示処理サブルーチン(図6を用いて後述する)を実行した(ステップS4)後に、本音色切替イベント処理を終了する。

【0064】ここで、ロード指示処理サブルーチンは、後述するように、転送速度TAに設定する値に応じて低速ロード指示と高速ロード指示とを1つのサブルーチンで切り換え可能のように構成されており、ステップS4では、転送速度TAに「低速」に対応する値を設定することにより低速ロード指示を行っている。これは、前述のように、音色切替時は、ロードを速く行う必要はないからである。

【0065】図5は、CPU3が実行するノートオンイベント処理の手順を示すフローチャートであり、上記制

御処理(1)を実現したものである。本処理は、演奏データとしてノートオンイベントが供給されたときに、起動される。

【0066】同図において、まず、供給されたノートオンイベントからノートナンバおよび(&)ベロシティを取得し、それぞれ、RAM5の所定位置に確保された領域NN(以下、この領域の内容を「ノートナンバNN」という)および領域VEL(以下、この領域の内容を「ベロシティVEL」という)に格納する(ステップS11)。

【0067】次に、ノートナンバNNおよび音色TC(この音色TCは、ノートオンイベントが供給されるときには既に設定されている)に対応した波形IDを検出する(ステップS12)。

【0068】そして、この検出された波形IDの波形データがキャッシュメモリ11上にロードされているかを判別する(ステップS13)。この判別は、前記ステップS3での判別と同様の方法によって行う。

【0069】ステップS13で、目的の波形データがキャッシュメモリ11上にロードされていないときは、前記ステップS4と同様にして、ロード指示処理サブルーチン(ただし、高速ロード指示)を実行した(ステップS14)後に、ステップS15に進む一方、目的の波形データがキャッシュメモリ11上にロードされているときは、ステップS14をスキップしてステップS15に進む。

【0070】ステップS15では、空きチャンネルを検出して、そのチャンネルに発音割り当てを行う。

【0071】続くステップS16では、割り当てたチャンネルに、音色TCおよびノートナンバNNに応じた各種パラメータを設定する。具体的には、前記レジスタ9の600ch分のchデータ格納領域中、割り当てチャンネルに対応するchデータ領域に、前記FナンバF、波形ID、相対アドレス、DCF制御情報、EG制御情報およびその他の情報を書き込む。

【0072】そして、当該チャンネルに対してノートオンを設定した(ステップS17)後に、本ノートオンイベント処理を終了する。ここで、ノートオンの設定は、上記ステップS16のchデータ領域中のノートオン/オフデータをノートオン(“1”)に設定することにより行う。

【0073】図6は、前記ステップS4およびS14のロード指示処理サブルーチンの詳細な手順を示すフローチャートである。本ロード指示処理によって、前記制御処理(5)が実現される。

【0074】同図において、まず、レジスタ9の600波形分のC管理データ格納領域中、転送ステートTSが“0”の領域cwk(kは、1~600のいずれかの整数値)のうち最小のkを取得する(ステップS21)。

【0075】次に、決定(または検出)された波形ID

が示す波形データを記憶するためのキャッシュメモリ11上の領域を割り当てる(ステップS22)。この領域割当は、たとえば、次のようにして行う。すなわち、

(1) キャッシュメモリ11に記憶されている複数の波形データを仮想アドレス上で前詰めする

(2) その後に形成された空き領域中、波形IDが示す波形データを書き込む領域を確保する

(3) 空き領域が足りない場合には、現在使用されていない波形データが記憶されている領域を開放し、上記

(1)および(2)の処理を繰り返す。

【0076】続くステップS23では、領域cwkに必要なパラメータを設定する。具体的には、前記波形ID、転送ステートTS、キャッシュ記憶位置仮想アドレス、トータルサイズおよび波形メモリ記憶位置リニアアドレスを書き込む。

【0077】そして、同時にロードする波形データの優先度に応じて、各波形データの転送速度TAを割り当て(分配)した(ステップS24)後に、本ロード指示処理を終了する。ここで、分配は、前述した方法によって行い、この分配された各転送速度TAは、当該領域cwk中、対応する転送速度TA領域に書き込まれる。これにより、前記音色切換イベント処理では、「低速」に相当する転送速度が設定され、前記ノートオンイベント処理では、「高速」に相当する転送速度が設定される。なお、「低速」および「高速」は、音色切換イベント処理およびノートオンイベント処理での転送速度の相対的な表現であって、分配の結果、ノートオンイベント処理での転送に、より大きな速度が割り当てられるという意味である。「低速」が設定された場合でも、他に転送が行われていなければ、転送速度TAに“120”全てが割り当てられる。

【0078】図7は、前記キャッシュ制御部12が実行する転送処理の手順を示すフローチャートであり、前記制御処理(3)および(4)を実現したものである。本転送処理は、上記図6のロード指示処理サブルーチンによってロード指示が行われると、起動される。

【0079】図7において、まず、ロードが指示(TS=1)された領域cwk中、最小のk(初期値)を検出し、RAM5の所定位置に確保された領域L(以下、この内容を「インデックスL」という)に格納する(ステップS31)。

【0080】そして、インデックスLが示す領域cwk内の転送速度TA(L)を取得して、RAM5の所定位置に確保されたソフトウェアカウンタ領域CNT(以下、この内容を「カウンタCNT」という)に格納する(ステップS32)。

【0081】次に、インデックスLが示す領域cwk内の波形ID(L)に対応する波形データを波形メモリ10から読み出して、前記ステップS22で割り当てられたキャッシュメモリ11上の領域に転送する(ステップ

S33)。この転送は、1または複数個のブロック分、すなわち、50ブロック/秒の転送速度が得られるだけのブロック分行われる。

【0082】続くステップS34では、波形ID(L)に対応する波形データの転送が完了したか否かを判別する。この判別は、領域cwL内の転送済みサイズとトータルサイズとを比較することによって行う。

【0083】ステップS34で、波形ID(L)に対応する波形データの転送がまだ完了していないときには、カウンタCNTを“1”だけデクリメントした(ステップS35)後に、カウンタCNTが“0”であるか否かを判別する(ステップS36)。

【0084】ステップS36で、CNT≠0のときには、転送すべき波形データのブロックがあるので、前記ステップS33に戻って、1または複数個のブロックの転送を繰り返す。

【0085】一方、ステップS34で、波形ID(L)に対応する波形データの転送が完了したときには、領域cwL内の転送ステートTS(L)に“2”を設定した(ステップS37)後に、転送ステートTSが“1”の領域cwkが他に存在するか否かを判別する(ステップS38)。

【0086】ステップS38で、転送ステートTSが“1”の領域cwkが他に存在しないときには、本転送処理を終了する一方、転送ステートTSが“1”の領域cwkが他に存在するときには、その領域cwk間で、転送速度TAの再配分を行う(ステップS39)。再配分は、たとえば、転送中の波形データ間でトリガ要因や優先度を考慮して行う。そして、このようにして再配分された転送速度TAを、対応する転送速度TA領域に書き込む。

【0087】続くステップS40では、次に転送すべき波形データを求めるために、転送ステートTSが“1”の領域cwk中の、前列に転送処理した領域cwLの次のk(ここで、次の“k”は、循環的に選択される。たとえば、転送ステートTSが“1”の領域がk=2, 5, 10, 25のとき、cwLの“L”が“2”であれば次の“k”は“5”であり、“L”が“25”であれば、次の“k”は“2”である。)を取得して、インデックスLに設定し、前記ステップS32と同様にして、インデックスLが示す領域cwL内の転送速度TA(L)を取得し、カウンタCNTに格納した(ステップS41)後に、前記ステップS33に戻る。

【0088】一方、ステップS36で、CNT=0のときには、転送すべき波形データのブロックがないので、上記ステップS40に進む。

【0089】このように、本実施の形態では、楽音を生成するときには、キャッシュメモリに記憶された波形データのみを使用して高速に波形データを供給するようにしたので、波形メモリのデータ転送速度の遅さに依存す

る発音チャンネル数の制限を解消することができる。

【0090】キャッシュヒット時には、キャッシュメモリに保持された波形データを用いて楽音を生成するので、波形メモリからキャッシュメモリへの波形データの転送を行う必要がない。

【0091】発音が指示された波形データのキャッシュミス時には、他の要因でロードが指示された波形データより高い優先度でキャッシュメモリに転送されるとともに、その転送速度の下限は、当該波形データのFナンバーに応じて制御されるので、発音の途中にキャッシュメモリに読み出すべき波形サンプルが未転送という状態を解消することができ、楽音を安定して再生することができる。

【0092】波形サンプルの読み出しアドレスの進行と、波形メモリからキャッシュメモリへの波形データの転送とが非同期でなされるので、キャッシュメモリへの波形データの転送方法が制限されず、たとえばブロック転送やバースト転送等最も効率のよい転送方法を使用することができる。

【0093】波形データのキャッシュメモリへのキャッシングは1波形単位で行われるので、すなわち、波形データを分割して、その一部のみをキャッシングするという方法ではないので、そのデータ管理が簡単になる。

【0094】音色選択に応じて優先度の高い波形データを予めキャッシュメモリに記憶するようにしたので、初めて使用する波形データについても効率よくキャッシュヒットさせることができる。

【0095】なお、本実施の形態では、波形データを記憶する記憶装置として波形メモリ(ROM)を使用したのが、これに限らず、ランダムアクセス可能なハードディスク装置やMD装置等を使用してもよい。この場合には、まず、全波形データの各先頭部のみをキャッシュメモリに記憶しておき、その後、使用される波形や必要な波形を当該記憶装置から順次キャッシュメモリに転送するようにする。

【0096】また、本実施の形態では、音色選択時に優先度の高い波形データをロードするようにしているが、これに限らず、演奏音域やテンポ等の変化に応じて、その後に使用されそうな波形データを予測し、この予測された波形データを予めロードするようにしてもよい。または、演奏者が今後使用したい波形データを、その使用時点で先立って指定し、この指定された波形データを予めロードするようにしてもよい。

【0097】さらに、本実施の形態では、波形メモリから読み出した波形データをキャッシュメモリに転送するとき、その転送場所は空いている任意の場所として、いわゆるフルアソシアティブ方式を採用したが、これに限らず、ダイレクトマップ方式またはセットアソシアティブ方式のいずれの方式を採用してもよい。

【0098】なお、上述した実施の形態の機能を実現す

るソフトウェアのプログラムコードを記録した記憶媒体を、システムまたは装置に供給し、そのシステムまたは装置のコンピュータ（またはCPU3やMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、本発明の目的が達成されることは云うまでもない。

【0099】この場合、記憶媒体から読出されたプログラムコード自体が本発明の新規な機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0100】プログラムコードを供給するための記憶媒体としては、たとえば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROM4などを用いることができる。また、他のMIDI機器や通信ネットワークを介してサーバコンピュータからプログラムコードが供給されるようにしてもよい。

【0101】また、コンピュータが読出したプログラムコードを実行することにより、上述した実施の形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOSなどが実際の処理の一部または全部を行い、その処理によって上述した実施の形態の機能が実現される場合も含まれることは云うまでもない。

【0102】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPU3などが実際の処理の一部または全部を行い、その処理によって上述した実施の形態の機能が実現される場合も含まれることは云うまでもない。

【0103】

【発明の効果】以上説明したように、請求項1または5に記載の発明によれば、読み出し指示に係る波形データが第2の記憶手段に記憶されているときには、その記憶されている波形データに基づいて、また、読み出し指示に係る波形データが第2の記憶手段に記憶されていないときには、その波形データを第1の記憶手段から読み出して、第2の記憶手段に転送した後、その記憶されている波形データに基づいて、楽音を生成するようにしたので、すなわち、楽音を生成するときには、データ転送速度が高速の第2の記憶手段に記憶された波形データのみを使用して高速に波形データを供給するようにしたので、データ転送速度が低速の第1の記憶手段に依存する発音チャンネル数の制限を解消することができる。

【0104】また、第1の記憶手段から第2の記憶手段への波形データの転送と、第2の記憶手段から読み出さ

れた波形データに基づく楽音の生成とが非同期で行われるので、第2の記憶手段への波形データの転送方法が制限されず、たとえばブロック転送やバースト転送等最も効率のよい転送方法を使用することができる。

【0105】請求項2に記載の発明によれば、読み出し指示に係る波形データが波形データ単位で第2の記憶手段に転送されるので、すなわち、波形データを分割して、その一部のみが第2の記憶手段に転送されるという方法ではないので、そのデータ管理が簡単になる。

10 【0106】請求項3または6に記載の発明によれば、音色切替が指示されたときに、該指示された音色において優先的に用いられる波形データが第2の記憶手段に記憶されていないときには、該波形データが第1の記憶手段から読み出されて、前記第2の記憶手段に転送されるので、初めて使用する波形データについても効率よくキャッシュヒットさせることができる。

20 【0107】請求項4または7に記載の発明によれば、発音指示に係る波形データが第2の記憶手段に記憶されていないときの波形データの各波形サンプルの転送速度を、第2の記憶手段から読み出された波形サンプルに基づく楽音生成時の各波形サンプルの読み出し速度より高速にしたので、発音の途中で第2の記憶手段に読み出すべき波形サンプルが未転送という状態を解消することができ、楽音を安定して再生することができる。

【図面の簡単な説明】

【図1】本発明の一実施の形態に係る楽音生成装置の概略構成を示すブロック図である。

【図2】図1のレジスタに格納されるchデータのデータフォーマットを示す図である。

30 【図3】図1のレジスタに格納されるC管理データのデータフォーマットを示す図である。

【図4】図1の楽音生成装置、特にCPUが実行する音色切替イベント処理の手順を示すフローチャートである。

【図5】図1のCPUが実行するノートオンイベント処理の手順を示すフローチャートである。

【図6】図4または図5のロード指示処理サブルーチンの詳細な手順を示すフローチャートである。

40 【図7】図1のキャッシュ制御部が実行する転送処理の手順を示すフローチャートである。

【符号の説明】

3 CPU（判別手段、転送手段、音色切替指示手段）

10 波形メモリ（第1の記憶手段）

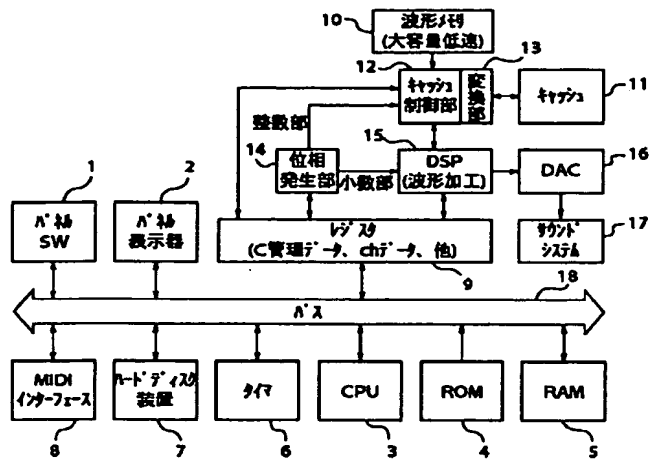
11 キャッシュメモリ（第2の記憶手段）

12 キャッシュ制御部（転送手段、楽音生成手段）

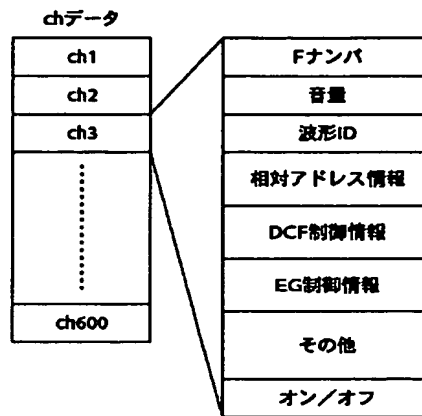
13 アドレス変換部（転送手段、楽音生成手段）

15 DSP（楽音生成手段）

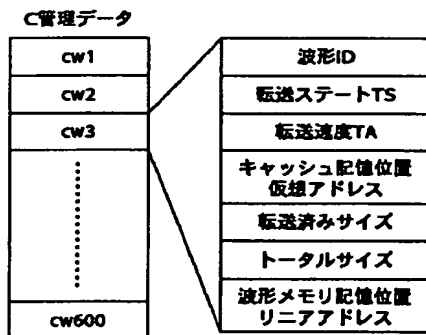
【図1】



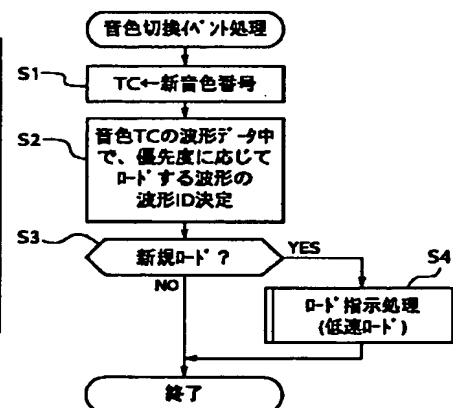
【図2】



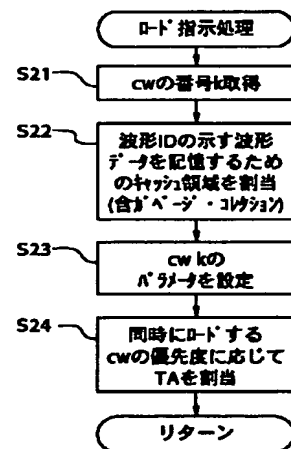
【図3】



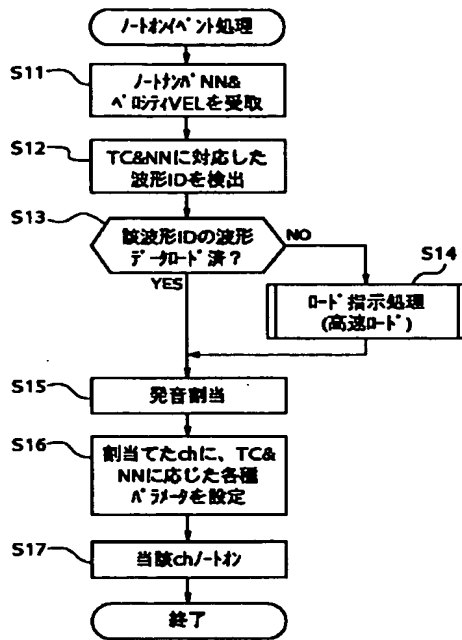
【図4】



【図6】



【図5】



【図7】

